FORE Systems PowerHub 6000 Installation and Configuration Manual, Software Version 6-2.6.3.0

PN 400-1464-0001, Rev C, Issue 1, July 1996

FORE Systems
174 Thorn Hill Road
Warrendale, PA 15086-7586
Phone: 412-772-6600
Fax: 412-772-6500
Product Information:
Phone: 1-888-404-0444
Fax: 412-635-6325
Technical Support:
1-800-671-FORE
Internet: info@fore.com
or http://www.fore.com

COMPLIANCE WITH EMISSIONS AND SAFETY STANDARDS:
This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC Rules. In addition, this product has been tested and found to be capable of operating within the specifications set forth under EN 55022 for Class A equipment. This product also complies with the provisions of EN 50082-1 relating to RFI, EMI, and ESD, which along with EN 60950 (see next paragraph) allows the product to carry the “CE” mark.

Tests performed by Inchcape Testing Services/ETL Test Laboratories demonstrate that this equipment meets UL 1950, CSA 950, and EN 60950 safety standards.

The emissions standards are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his or her own expense.

PUBLICATION HISTORY

PowerHub 6000 Installation and Configuration Manual, Software Version 6-2.6.3.0

PN 400-1464-0001

Date	Rev	Issue	Description
June, 1995	A	1	First edition, corresponding to system software version 2.6-6 and first release of 6000 architecture.
August, 1995	B	1	Second edition, corresponding to system software version 2.6-6.
July, 1996	C	1	Third edition, corresponding to system software version 6-2.6.3.0.

ABOUT THIS BOOK

This book is for system administrators or others responsible for installing and configuring the PowerHub 6000. This book describes the PowerHub 6000 hardware, as well as how to install it and how to upgrade it. In addition, this book describes the user interface and the commands you use to configure the PowerHub hardware and display configuration information.

This book is divided into the following parts:

Part 1: Overview	Briefly describes the configuration features documented in this book, then describes the PowerHub hardware and the software files shipped with the hub.
Part 2: Installation	Contains procedures for installing the hub, changing or upgrading hub components, and installing software upgrades.
Part 3: Configuration	Describes the commands you use to configure the hardware and user interface, create environment and configuration files, and perform other configuration tasks. Also describes the Packet Engine boot PROM commands.

Part 4: Appendices	Describes how to run the diagnostic self-tests and provides information about various netbooting options. Also lists the pinouts of the 10Base-T and Fast Ethernet connectors used on the PowerHub 6000.

OTHER BOOKS

You can find additional information about the PowerHub 6000 in the following books:

PowerHub Software Manual, V 2.6 (Rev C)
Describes the commands to configure the PowerHub 6000 for bridging and for IP routing. This manual also describes how to configure the PowerHub system to perform RIP routing and IP Multicasting.

PowerHub Supplementary Protocols Manual, V 2.6 (Rev C)
Describes the commands to configure the hub for AppleTalk, DECnet, and IPX routing, as well as information on implementing IP Security.

PowerHub OSPF Addendum (Rev A or later)
Describes the PowerHub implementation of the OSPF (Open Shortest Path First) routing protocol and how to configure your PowerHub system as an OSPF router.

PowerHub 6000 FDDI Addendum (Rev A or later)
Describes the PowerHub 6000 FDDI daughter cards and how to install them and configure them.

TYPOGRAPHICAL CONVENTIONS

The following typographical conventions are used in this manual:

This type style...	Indicates...
AaBbCcDd	A term that is being defined. Example: The *Packet Engine* is the PowerHub 6000's centralized packet processing and forwarding engine.
`AaBbCcDd`	A command name. PowerHub commands are case sensitive; they should always be entered as shown in the manual and on-line help. Example: **`showsmtmib`**
`|`	1) Separates the full and terse forms of a command or argument: • The full form is shown on the left of the **`|`**. • The terse form is shown on the right of the **`|`**. When you type the command or argument, you can type either the full form or the terse form. Example: **`showsmtmib|ssm`** In this example, you can type **`showsmtmib`** or **`ssm`**. 2) Separates mutually exclusive command arguments. Example: **`set stp enl|dis`** In this example, the command **`set stp`** can accept either **`enl`** or **`dis`**, but not both.
`[ ]`	Enclose optional command arguments or options. Example: **`setuser [root|monitor]`** In this example, the **`[ ]`** enclose an optional argument. You can issue the command without the argument(s) shown in **`[ ]`**. However, if specified, the argument must be one of the two options listed between the **`[ ]`**.

<AaBbCcDd> Indicates a parameter for which you or the PowerHub 6000 supplies a value. When used in command syntax, *<italics>* indicates a value you supply. Example:

showfile *<file-name>*

In this example, *<file-name>* is a parameter for which you must supply a value when you issue this command.

`AaBbCcDd` Indicates a field name or a file name.

A field name example:

When you boot the PowerHub software, the `login:` prompt is displayed.

A file name example:

When you boot the PowerHub software, the system looks for a file named `cfg`.

AaBbCcDd or **AaBbCcDd**

Indicates text (commands) displayed by the PowerHub software or typed at the command prompt. To make typed input easy to distinguish from command prompts and output, the typed input is shown in darker type. Example:

```
1:PowerHub# pm view all 10 on 12
Port 10 (all) being viewed on: 12
```

In this example, the user types "**pm view all 10 on 12**" and the software responds "`Port 10 (all) being viewed on: 12`".

Contents

Part 1: Overview

Part 1: Overview

This part describes the PowerHub 6000 hardware, the software files, and how slots and segments are numbered in the PowerHub chassis.

- If you want to skip this part and install your PowerHub 6000 now, go to Part 2: Installation.
- If your PowerHub 6000 is installed and you are ready to use software commands to configure it, go to Part 3: Configuration.

This part contains the following chapters:

1 Features Overview

This chapter gives an overview of the hub configuration and management features provided by the PowerHub 6000 hardware and software. The features introduced in this chapter are fully described in other manuals or in other chapters in this manual. The sections in this chapter tell which chapters or sections to reference for more information about the features.

This chapter also defines the terms “slot,” “port,” and “segment,” and shows how they apply to the PowerHub 6000.

1.1 HARDWARE

The following sections describe the major hardware features of the PowerHub 6000:

- Chassis configuration options.
- Full-duplex segments.
- Redundant power and load sharing.
- Multiple boot sources.

1.1.1 Chassis Configuration Options

The PowerHub 6000 chassis is designed for easy reconfiguration in response to your networking needs. You do not need to send the hub to the factory to make hardware changes. You can perform the upgrades using a few standard tools (usually ordinary screwdrivers). Your network is down for just a few minutes while you perform the upgrades.

> **NOTE:** The PowerHub 6000 does not support “hot-swapping.” To make hardware changes, you must power down the system.

1.1.1.1 *Network Segments*

The basic chassis provides twelve 10Base-T Ethernet segments, but you can expand the chassis to include the following connection options, in addition to the twelve 10Base-T segments:

- Six or twelve 10Base-FL (FOIRL-compatible) Ethernet segments.
- Twelve or 24 additional 10Base-T Ethernet segments on RJ-45 connectors.
- 12 or 24 additional 10Base-T Ethernet segments on Champ connectors.
- One or two 100 Mb/s Ethernet segments (100Base-TX, 100Base-FX, or 100Base-T4).
- One FDDI DAS, for multimode or single-mode fiber.

In addition, if you need Ethernet connection types other than 10Base-T, you can bypass from one to six of the 10Base-T connectors on the Packet Engine with any combination of the following types of Ethernet connections:

- 10Base-FB
- 10Base-FL (FOIRL-compatible)
- AUI
- BNC

NOTES: The PowerHub 6000 is designed to handle the maximum valid lengths of Ethernet and FDDI packets. However, packets larger than 1518 bytes (the maximum Ethernet length) that either are broadcast packets or are addressed directly to the hub itself may cause unexpected results. Other invalid packets are dropped by the hub.

For Novell IPX version 4.0 and later, you must set the MTU size equal to or less than 1500 bytes because the IPX specification provides no means for fragmentation.

High collision rates can occur on Ethernet segments that are transmitting bursty FDDI-originated traffic. These high collision rates are normal for this type of traffic and do not indicate a problem with the PowerHub system.

1.1.1.2 *Performance Enhancing Options*

Depending on your network needs, you can add either of the following performance enhancing options:

Flash Memory Module (if you ordered your hub without it)
: The Flash Memory Module is very useful for fast system booting, while it allows you to store and manage files locally on the hub.

Packet Accelerator
: The Packet Accelerator increases the packets-per-second throughput of the hub by adding RISC CPUs to increase the processing power of the Packet Engine. The Packet Accelerator is required in systems that use more than 34 segments.

To accommodate software versions or features that require more DRAM memory, the Packet Engine contains a standard DRAM SIMM socket to make DRAM upgrades simple.

Chapter 5 contains complete instructions for adding these hardware options.

1.1.2 Redundant Power and Load Sharing

The chassis allows installation of a second (redundant) power supply to ensure uninterrupted operation of the hub in the unlikely event that a power supply fails. Moreover, when redundant power supplies are installed, all the power supplies load share, thereby extending the life span of each supply and further reducing the chance of power supply failure.

1.1.3 Full-Duplex Segments

The PowerHub connections for 10Base-T, 10Base-FL, 100Base-FX and 100Base-TX segments can be configured to operate in full-duplex mode. Full-duplex mode doubles the maximum bandwidth on these segments. For the 10Base-T and 10Base-FL segments, whose theoretical maximum bandwidth is 10 M/bs, full-duplex mode increases the maximum bandwidth to 20 Mb/s. For Fast Ethernet segments, full-duplex doubles the maximum bandwidth from 100 Mb/s to 200 Mb/s.

Note that Ethernet segments running in full-duplex mode are not subject to collisions.

1.1.4 Lock Switch and Password Protection

The PowerHub 6000 provides security at different levels. On the hardware level, the Packet Engine contains a Lock Switch jumper. When this jumper is set to Locked, the software requires a valid login and user ID for access to the user interface.

The Lock Switch is set to Unlocked at the factory, but can be reset by removing a hardware jumper on the Packet Engine. For information about setting the Lock Switch jumper to Locked, see Section 5.12 on page 103.

In addition, some chassis contain an external Lock Switch. The *Lock Switch* performs the same function as the Lock Switch jumper, but is accessible from the chassis front panel, next to the reset (RST) button. When the Lock Switch is set to the unlocked position (U), anyone capable of establishing an RS-232 or TELNET connection to the hub can access the user interface and begin a management session on the hub. When the Lock Switch is in the locked position (L), users must enter a login name and password before they can begin a session and issue commands.

You can override the external Lock Switch by setting the Lock Switch jumper.

For information about logical filters and other security in the software, see Section 1.2.10.

1.1.5 LEDs

At any time, you can get status information for the Packet Engine or network segments by directly observing LEDs on the hub.

The Packet Engine has status LEDs. The status LEDs indicate whether the module is booting or is performing normal runtime tasks. If unexpected events cause the Packet Engine to crash and reboot, a “sticky” Alarm LED indicates that the crash has occurred.

In addition, configurable traffic LEDs on the Ethernet segments indicate the traffic activity on each segment. You can configure the traffic LEDs to show network activity and collisions, or send and receive traffic (but not collisions).

For more information about the LEDS, see the following sources:

- Packet Engine LEDs, Section 2.3.1 on page 25.
- Ethernet traffic LEDs, Section 2.3.11 on page 29.
- FDDI traffic LEDs, Section 2.3.12.2 on page 31.

1.1.6 Multiple Boot Sources

Depending on your PowerHub configuration and network configuration, you can configure the hub to use one the following boot sources:

- Flash Memory Module (if installed).
- BOOTP/TFTP server.

Moreover, you can configure the hub to try both of these sources, in either order. If one method fails for some reason, the hub still has an opportunity to boot successfully using the other boot source. For example, in an installation containing many PowerHub Intelligent Switching Hubs, you could configure the hubs to boot over the network, as depicted in Figure 1–1. In such a configuration, all the hubs download and boot from the same system software, while each hub’s unique configuration file is downloaded from the server.

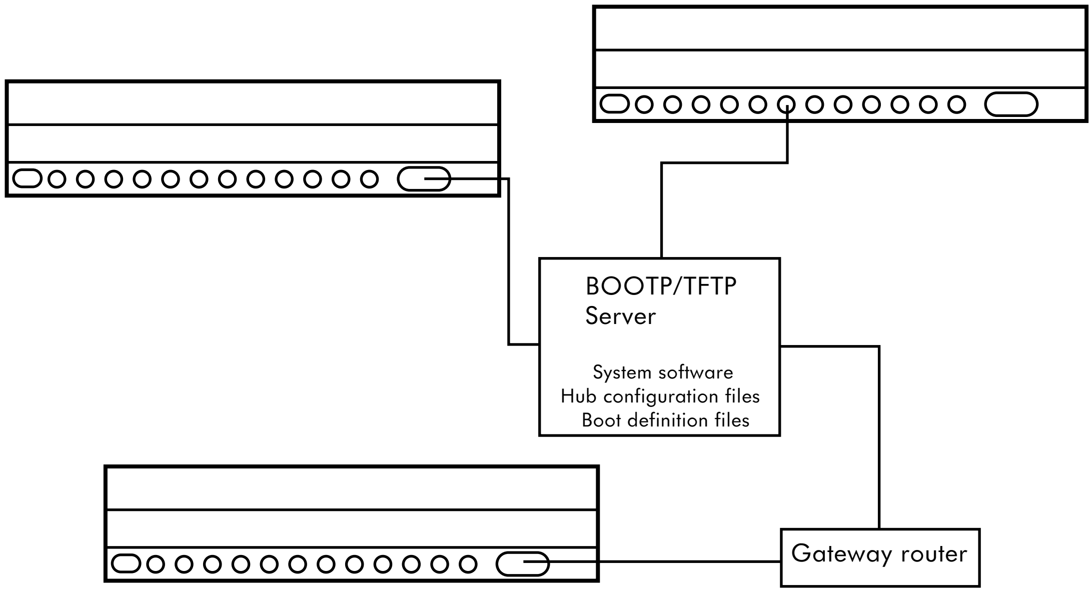


FIGURE 1–1 Netbooting multiple hubs from the same server.

For example, you could configure the hubs depicted in Figure 1–1 to boot from the BOOTP/TFTP server. In a case where network problems prevent a hub from netbooting, that hub tries the secondary boot source (Flash Memory Module).

Notice that you can configure the PowerHub 6000 to netboot even when the hub and boot server are separated by a gateway. The PowerHub software contains a feature called *IP Helper* that makes this type of netbooting easy to configure. (See Section B.2.2 on page 234 in this manual and Chapter 5 in the *PowerHub Software Manual, V 2.6* (Rev C).)

Section 4.3.4 on page 60 contains instructions for configuring the boot source. For information on netbooting, see Appendix B.

1.2 SOFTWARE

The following sections describe the major software features of the PowerHub 6000:

- DOS-like file management system.
- Up to three concurrent command-line sessions.
- Easily created and loaded system and session parameter files.
- On-line hardware information.
- Bridging and multi-protocol routing.
- OSPF (Open Shortest Path First) routing protocol.

- RIP (Routing Information Protocol).
- IP Multicasting.
- Virtual LANs.
- Security filters.
- Statistics for segment traffic.
- Automatic segment-state detection.
- Traffic monitoring.
- SNMP and MIB support.

These features are described in the following sections.

1.2.1 DOS-Like File Management System

The Flash Memory Module uses a DOS-like file management system that lets you list, display, copy, rename, and remove files. Files created or stored in the Flash Memory Module can be edited using a standard ASCII editor on another device (PC, Macintosh, and so on). Note that the PowerHub file management system does not support hierarchical directory structures.

1.2.2 Concurrent Command-Line Sessions

The PowerHub 6000 supports up to three concurrent command-line sessions. The primary command-line session is always established through a direct connection between a PC or modem and an RS-232 port (labeled TTY1) on the PowerHub chassis.

In addition, you can use TELNET to establish one or two in-band sessions with the command-line interface. To use TELNET to access the hub, you first need to define an IP interface on the segment attaching the PowerHub 6000 to your terminal.

> **NOTE:** The Packet Engine contains a second TTY port (TTY2) used only for diagnostic purposes. This port is not accessible from the Packet Engine front panel. Do not attempt to use TTY2 unless advised to do so by FORE Systems TAC.

As shown in Figure 1–2, you can have one directly-attached command-line session and two TELNET sessions to the command-line interface.

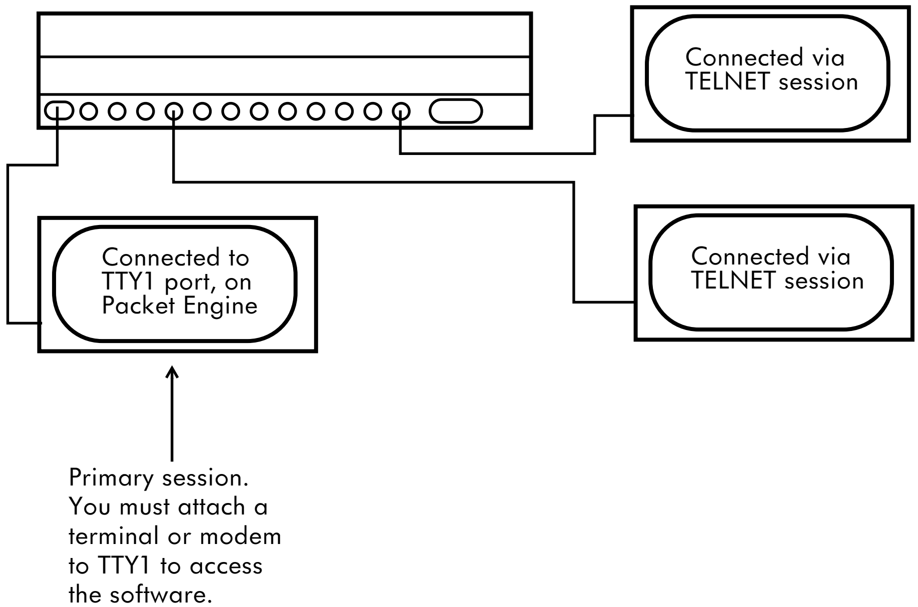


FIGURE 1–2 Three concurrent command-line sessions.

Each command-line session supports features, such as command histories, command aliases, and timed commands. See Chapter 7 and Section 8.10 on page 149 for information about these features.

1.2.3 Configuration and Environment Files

When you configure your PowerHub 6000, you define parameters such as the name, routing interface definitions, and filters. The hub does not retain these definitions across power cycles. However, you can save these definitions and reload them at any time using PowerHub software commands.

The **`mgmt savecfg`** `cfg` command, for example, creates a file called `cfg` and saves it on the Flash Memory Module (if present). This file contains ordinary PowerHub commands, such as you might issue to configure the system. When you boot the hub, the `cfg` file is read, the commands are executed, and the hub is configured according to your specifications.

You can create multiple configurations and load them as needed. See Section 9.8 on page 183 for more information about configuration files.

Parameters for command-line sessions also can be stored and read into the hub. Session parameters are stored in environment files. Like configuration files, environment files contain ordinary PowerHub commands, and can be loaded during log in. Unlike configuration files, environment files apply only to the command-line session in which they are loaded, not to the entire hub. See Section 8.10 on page 149 for more information about environment files.

1.2.4 Easily Accessible Diagnostic Files

The PowerHub 6000 is designed for reliable operation under diverse traffic demands. However, if the PowerHub 6000 crashes for any reason, it attempts to reboot and write a dump file to the floppy drive or the Flash Memory Module. This dump file is always named `fore.dmp` and contains information used by FORE Systems TAC (Technical Assistance Center) to help diagnose the cause of the system crash. See Section 3.2 on page 44 for more information.

FORE Systems TAC also can use the contents of the `dispcfg` file to diagnose system problems. The `dispcfg` file is a configuration file provided on the system software diskettes. When you read the file (by issuing the **`mgmt readcfg dispcfg`** command), the system configuration is displayed on the management terminal.
(See Section 9.8.3 on page 190.)

1.2.5 On-Line Hardware Information

At any time during normal hub operation, you can use PowerHub software commands to display the following hardware information:

- Chassis configuration.
- Power supply status.
- Identification information and power requirements for the Packet Engine and the NIMs (Network Interface Modules).
- The temperature of the Packet Engine.
- Segment state, status, and statistics.
- Termination state of BNC segments (terminated or unterminated).

Figure 1–3 shows an example of one type of information you can display for the hardware. In this example, the current temperature of the Packet Engine is displayed.

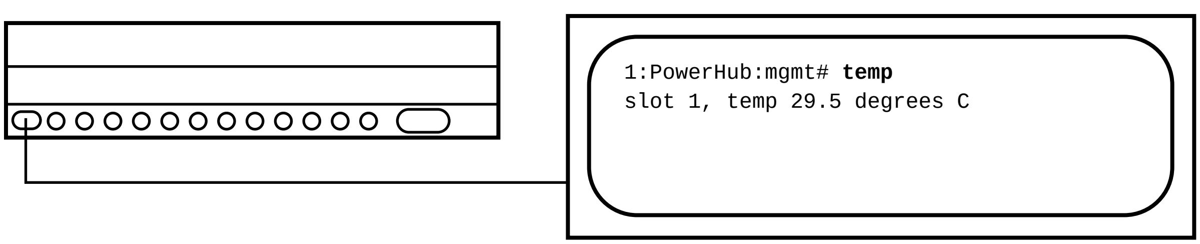


FIGURE 1–3 Example of on-line chassis information.

1.2.6 Bridging and Multi-Protocol Routing

You can configure PowerHub segments to bridge (as specified in IEEE 802.1d) as well as route packets according to any combination of the following standard protocols:

- IP
- IPX
- AppleTalk
- DECnet

For networks containing both Ethernet and FDDI segments, you can use IPX encapsulation bridging to bridge frames forwarded between Ethernet and FDDI.

In addition, the PowerHub software contains an implementation of IP Multicasting, which lets you perform multicast routing for bandwidth-intensive applications, such as video conferencing.

The same segments can be used to bridge and route packets. In fact, you can configure a segment to not only bridge, but route all four protocols, and even to perform IP Multicasting.

For information on configuring segments for bridging and routing, see the following:

- IEEE 802.1d bridging — Chapter 2 in the *PowerHub Software Manual, V 2.6* (Rev C).
- IPX translation bridging — Chapter 2 in the *PowerHub Software Manual, V 2.6* (Rev C).
- IP routing — Chapter 5 in the *PowerHub Software Manual, V 2.6* (Rev C).
- IPX routing — Chapter 2 in the *PowerHub Supplementary Protocols Manual*, *V 2.6* (Rev C).
- AppleTalk routing — Chapter 1 in the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).
- DECnet routing — Chapter 3 in the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).
- IP Multicasting — Chapter 6 in the *PowerHub Software Manual, V 2.6* (Rev C).

1.2.7 OSPF Support

OSPF (Open Shortest Path First) is a routing protocol that enables each participating router to use a topological map of the network to route packets. OSPF routers exchange route information using *LSAs (link-state advertisements)*. An LSA is a packet that reports the link states (up or down) of a router's interfaces that are attached to devices in the OSPF network.

OSPF is an IGP (Interior Gateway Protocol). That is, OSPF distributes routing information within a single Autonomous System.

RIP (Routing Information Protocol) is another popular IGP used in TCP/IP networks, and also is implemented in the PowerHub software. RIP uses *distance vectors* to build route tables in RIP routers. See Section 1.2.8 for information about RIP.

OSPF does not use distance vectors for routes. Instead, OSPF uses the link states of the routers in the network to derive a picture of the network topology.

OSPF lets you logically group your networks into areas. By grouping your networks into areas, you can reduce the amount of routing traffic among OSPF routers. An area's topology is summarized before being sent to other areas, thereby reducing routing traffic.

For information about the PowerHub OSPF software, see the *PowerHub OSPF Addendum*.

1.2.8 RIP Support

PowerHub software version 6-2.6.3.0 supports RIP (Routing Information Protocol). RIP uses a distance-vector algorithm to calculate routes based upon the number of "hops" (intervening routers) to a node.

In addition, the PowerHub software contains an implementation of the IPX RIP and SAP (Service Advertisement Protocol) protocols.

For information on configuring segments for IP RIP, IPX RIP, or IPX SAP, see:

- IP RIP — Chapter 7 in the *PowerHub Software Manual, V 2.6* (Rev C).
- IPX RIP and SAP — Chapter 2 and in the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).

1.2.9 Virtual LANs

To make managing your network segments even easier, the PowerHub software lets you create VLANs (Virtual LANs). A VLAN is a network that spans two or more physical segments. The PowerHub software bridges, rather than routes, packets among segments in a virtual LAN. Figure 1–4 shows an example of a VLAN.

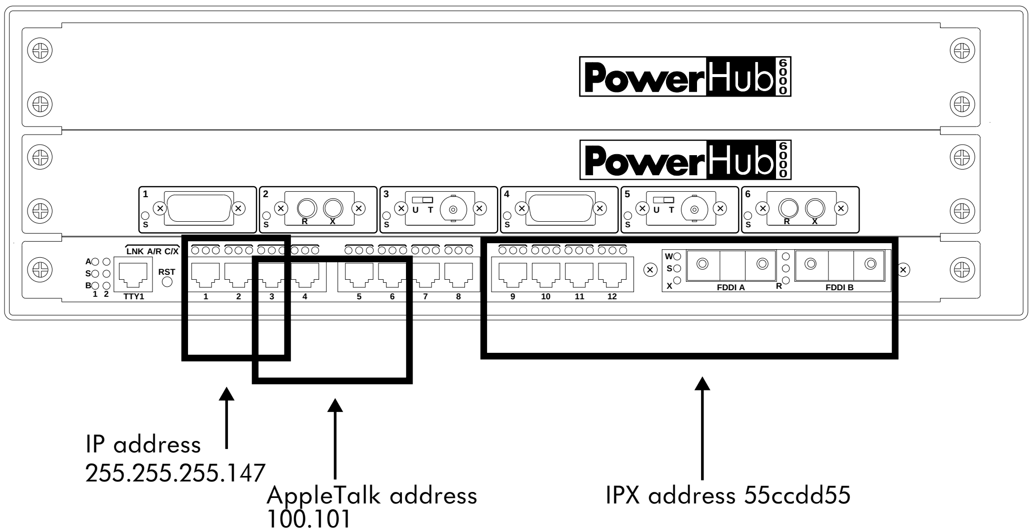


FIGURE 1–4 Examples of Virtual LANs.

As shown in Figure 1–4, a VLAN is a group of segments. Each segment in the VLAN has the same interface address. You can create VLANs for IP, IPX, and AppleTalk interfaces. Notice that VLANs can overlap. See Appendix D in the *PowerHub Software Manual, V 2.6* (Rev C) for more information about VLAN configuration.

1.2.10 Security Filters

In addition to the Lock Switch and password protection described in Section 1.1.4, you can define logical filters to control traffic entering and leaving the entire hub, or even specific segments. You can define filters on the following levels:

- Bridge
- TCP and UDP
- AppleTalk
- IP
- IP RIP
- IPX
- IPX RIP and SAP

The PowerHub software also contains an implementation of IP security as specified in RFC 1108. (See Chapter 4 in the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).)

For information on filters, see:

- Bridge — Chapter 9 in the *PowerHub Software Manual, V 2.6* (Rev C).
- TCP and UDP — Chapter 10 in the *PowerHub Software Manual, V 2.6* (Rev C).
- AppleTalk — Chapter 5 in the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).
- IP — Chapter 11 in the *PowerHub Software Manual, V 2.6* (Rev C).
- IP RIP — Chapter 12 in the *PowerHub Software Manual, V 2.6* (Rev C).
- IPX RIP and SAP — Chapter 6 in the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).

1.2.11 Segment and Packet Statistics

Using software commands, you can display packet statistics for any group or range of segments. For example, you can issue commands to display how many packets have been received by a particular segment since the software was booted.

For most statistics, two separate counters are maintained by the software:

Count since last system reset

This counter begins when the software is booted and continues until the hub is powered down or the software is rebooted. This counter always shows the count accumulated since the system was reset.

Count since last clear

This counter also begins when the software is booted, but can be cleared at any time using a software command. When you display the contents of this counter, the count displayed is the count subsequent to the last time the counter was cleared. This type of counter immediately begins again at zero when you clear it.

For information about displaying segment packet statistics, or statistics for packet types associated with various routing protocols, see:

- Bridge statistics — Chapter 2 in the *PowerHub Software Manual, V 2.6* (Rev C).
- TCP statistics — Chapter 3 in the *PowerHub Software Manual, V 2.6* (Rev C).

• IP statistics	Chapter 5 in the *PowerHub Software Manual, V 2.6* (Rev C).
• IP Multicast statistics	Chapter 6 in the *PowerHub Software Manual, V 2.6* (Rev C).
• OSPF statistics	Chapter 3 in the *PowerHub OSPF Addendum*.
• IP RIP statistics	Chapter 7 in the *PowerHub Software Manual, V 2.6* (Rev C).
• SNMP statistics	Chapter 8 in the *PowerHub Software Manual, V 2.6* (Rev C).
• IPX statistics	Chapter 2 in the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).
• DECnet statistics	Chapter 3 in the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).
• AppleTalk statistics	Chapter 1 in the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).

1.2.12 Automatic Segment-State Detection

Using *automatic segment-state detection*, the PowerHub software can detect when the state (up or down) of a segment attached to the hub changes. Moreover, when this feature detects a state change, it automatically enables or disables bridging and routing on the changed segment and marks the change in table displays.

For example, the dark nodes in Figure 1–5 represent devices that have been powered off, failed, or lost connectivity to the PowerHub 6000 for some reason.

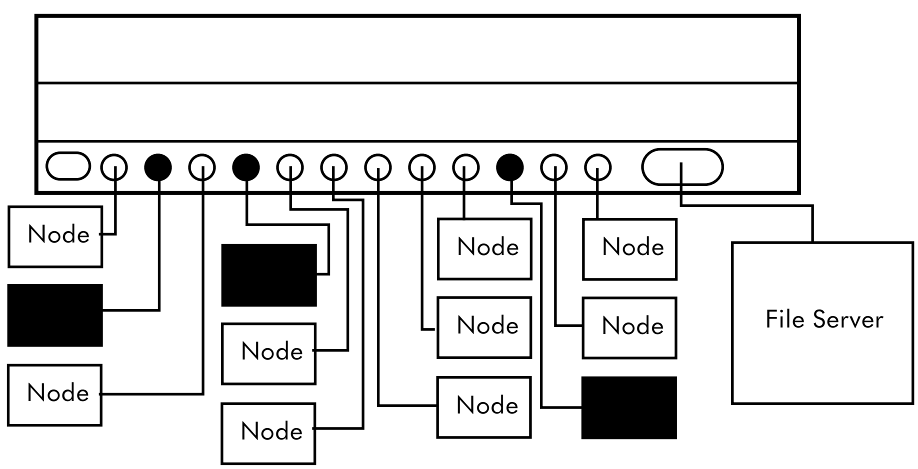


FIGURE 1–5 Automatic segment-state detection feature.

The automatic segment-state detection feature detects that these segments are no longer active and disables bridging and routing on the segments, while also marking the segments as down in the various statistics displays and data tables. This feature prevents the software from wasting packet processing resources with unnecessary attempts to forward traffic to the down segments.

See Section 9.5.4 on page 167 for information about using the automatic segment-state detection feature.

1.2.13 Port Monitoring

The Port Monitoring feature lets you monitor different types of traffic on any combination of PowerHub segments. You can configure the software to monitor packet traffic received on, forwarded from, or generated by specific segments. You even can monitor traffic between a pair of segments.

To use Port Monitoring, you need just one protocol analyzer (such as a Sniffer, LANalyzer, or Network Pharaoh), plugged into a single PowerHub segment. (See Figure 1–6.)

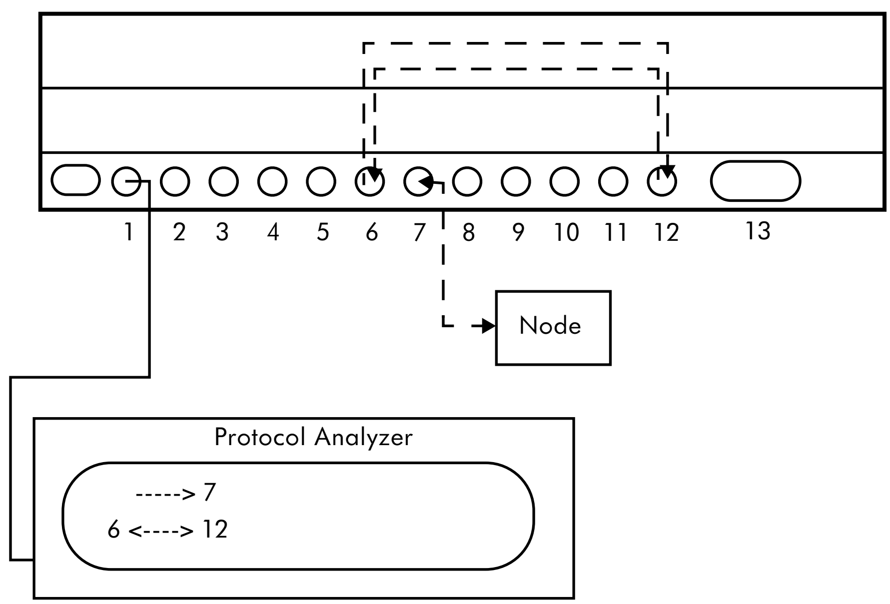


FIGURE 1–6 Port Monitoring feature.

Using software commands, you specify the segments and the types of traffic you want to monitor. In this example, Port Monitoring is configured to show traffic forwarded onto segment 7, and to show traffic between a particular pair of segments, 6 and 12. Segment 1 is being used as the monitoring segment, and is not being used for normal network traffic during monitoring.

To change what you are monitoring, you never need to move the protocol analyzer from the monitoring segment. You simply use software commands to change the monitoring configuration.

See Section 9.9 on page 193 for more information about Port Monitoring.

1.2.14 SNMP and MIB Support

You can use PowerHub software commands to define SNMP (Simple Network Management Protocol) managers and communities. Once you configure the hub so that it can be managed using SNMP, you can access the PowerHub implementations of the following standard MIBs (Management Information Bases):

- MIB II (RFC 1213).
- AppleTalk MIB (RFC 1243).
- Bridge MIB (RFC 1286).
- Ethernet MIB, Version II (RFC 1398).
- FDDI MIB (RFC 1512).
- OSPF V2 MIB (RFC 1253).

In addition, you can use SNMP to access objects in the PowerHub MIB, which is designated as enterprise MIB 390. The PowerHub MIB contains objects not contained by the standard MIBs listed above. This includes objects for displaying the software installed on the hub, as well as objects that display hardware configuration information about the PowerHub 6000.

See Chapter 8 in the *PowerHub Software Manual, V 2.6* (Rev C) for information about configuring the PowerHub 6000 for SNMP. For information about the standard and PowerHub MIB objects implemented on the hub, see Appendix A and Appendix B in the same manual.

1.3 SLOTS, SEGMENTS, AND PORTS

This manual, and the other PowerHub 6000 user documentation, uses the term “segment” to refer to a single 10 Mb/s or 100 Mb/s Ethernet collision domain or 100 Mb/s FDDI token-passing domain. For example, each physical network connection on the Packet Engine or other module in the PowerHub 6000 chassis connects to an individual segment. The term segment also applies when an Ethernet or FDDI segment is operating in full-duplex mode. Although no collisions occur on an Ethernet segment operating in full-duplex mode, the 20 Mb/s or 200 Mb/s Ethernet domain is still considered a segment.

The user interface uses the term “port” to refer to the segment connectors on the PowerHub 6000. In general, the terms “segment” and “port” are interchangeable on the PowerHub 6000. Each denotes one physical connection to a network segment.[1]

The word “port” also is used to refer to the TTY1 port on the Packet Engine. However, this port is always clearly distinguished in the text from microsegmented ports.

1. The one exception is DAS FDDI segments, which are described in this manual as containing multiple A and B “ports.”

1.3.1 How Slots and Segments are Numbered

The PowerHub 6000 chassis is divided into four slots:

- Packet Engine UTP segments (slot 1).
- Packet Engine fast segment(s) (slot 2). The fast segments can be Fast Ethernet or FDDI.
- Middle slot, which can contain the *UMM (Universal Media Module)* or a *NIM (Network Interface Module)* (slot 3).

  The UMM is an optional module you can install to bypass the first six UTP segments on the Packet Engine with six Ethernet segments of different media types, such as BNC, 10Base-FB, 10Base-FL, and AUI.

  A NIM is a plug-in module that provides additional segment connections to the hub. You also can install a NIM in Slot 3, rather than the UMM.

  The UMM and NIMs are described in Section 2.5 on page 36 and Section 2.4 on page 33.
- Top slot, which can contain a NIM (slot 4).

Each slot is managed separately by the software and shown separately by various software commands. Figure 1–7 shows the locations of the four slots in the PowerHub 6000 chassis.

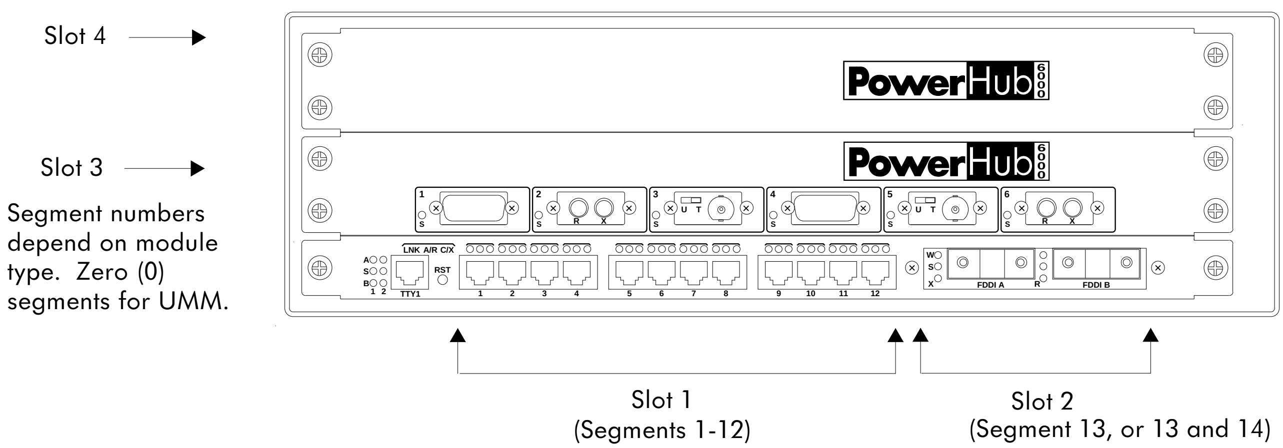


FIGURE 1–7 Slot locations in PowerHub 6000 chassis.

As shown in Figure 1–7, the UTP segments on the Packet Engine are always numbered 1 through 12, from left to right. The optional 100 Mb/s segment is number 13. (If you install the 2-segment version of a Fast Ethernet daughter card, slot 2 contains segments 13 and 14.)

In this example configuration, the UMM is used to bypass segment connectors 1 through 6 on the Packet Engine. However, because the UMM merely provides a different physical interface for the segments, they are still numbered 1 through 6. Accordingly, no segments are numbered in the middle slot, where the UMM is installed.

In this example, slot 4 is empty, so no segments are numbered for that slot. Segment numbers are added to slots 3 and 4 when you install NIMs in these slots. The number of segments for each slot depends upon the number of segments on the NIM.

1.3.1.1 Overriding Segment Numbers

When you power on the PowerHub 6000, the system numbers the segments according to the configuration of the chassis. For example, if the chassis contains an FDDI daughter card, the system numbers the segment in slot 2 as 13. The first segment in the NIM in slot 3 (if present) is number 14. If the chassis contains a 100 Mb/s Fast Ethernet daughter card, the system numbers the segments in slot 2 as 13 and 14 and numbers the leftmost segment in the NIM in slot 3 (if present) as 15.

By default, the system always numbers the segments sequentially from left to right, bottom to top. No “empty” segments (segment numbers for which there is no corresponding connector) are used.

In general, you do not need to worry about the segment numbering. However, if you need to override the default segment numbering for some reason, you can do so using the **`nvram set slotsegs`** command. This command lets you specify how many segments are valid for a slot in the chassis. If you change the configuration of the hub in such a way that segment numbers will be changed (ex: exchange the FDDI daughter card for a Fast Ethernet daughter card), you might want to override the segment numbering in order to use configuration files created before you changed the configuration of the chassis.

See Section 11.4 on page 214 for information on overriding the segment numbering.

1.3.2 Displaying Segment Numbers

You can display the segment numbers in your PowerHub 6000 by issuing the **`mgmt showcfg`** command. (See Section 9.3.6 on page 160.) This command shows basic configuration information for the chassis, including:

- The baud rates for the TTY (RS-232) ports.
- The slot that contains the Packet Engine. This is always slot 1.
- The presence and status of power supplies.
- The number of segments present in each slot in the chassis, and the medium type in use in each segment position.

Here is an example of the display produced by this command:

```
8:PowerHub:main# mgmt showcfg

Accelerator board is present. Accelerator IOP is being used.
Installed DRAM Size: 16 MB
tty1:  not set - using 9600 baud
tty2:  not set - using 9600 baud
PE:    slot 1
PM:    Good
  02/13 MM/MM
  01/01 AUI     FIBER   BNCT    AUI     BNCT    FIBER
        UTP     UTP     UTP     UTP     UTP     UTP
```

This example shows information for the PowerHub 6000 shown in Figure 1–7 on page 18. According to this display:

- The Packet Engine has the Packet Accelerator.
- The Packet Engine contains 16 MB of DRAM main memory.
- The baud rate on each TTY port is 9600.
- The Packet Engine is in slot 1.
- The power supply (or supplies) are “good”; that is, they are functioning normally.
- Two of the four slots in the chassis contain segments. (In this example, the slot information is shown in bold type.) For each slot, the slot number and the beginning segment number are shown. Following these are descriptions of the media types in use in the slot.

  Only slots 1 and 2 in this chassis contain segments. Slot 3 contains the UMM, but recall that the Ethernet electronics for the segment connections in the UMM are actually on the Packet Engine, in slot 1. Consequently, slot 3 is considered to be empty.

  The first row of information for slot 1 lists the Ethernet media types in use on segments 1 through 6. In this example, these are the EMAs (Ethernet Media Adapters) installed in the UMM. The second row for slot 1 shows the media type in use on segments 7 through 12. These are always UTP.

  The information for slot 2 shows that a multimode FDDI daughter card is installed. (For each of the FDDI ports in the module, the code `MM` indicates multimode.)

2 Hardware

This chapter describes the hardware architecture of the PowerHub 6000 Intelligent Switching Hub.

The basic PowerHub 6000 provides connection to twelve independent 10Base-T network segments, but can be expanded to connect to additional 10 Mb/s Ethernet segments, as well as 100 Mb/s Fast Ethernet and FDDI segments.

Figure 2–1 shows the front view of one possible configuration of the PowerHub 6000.

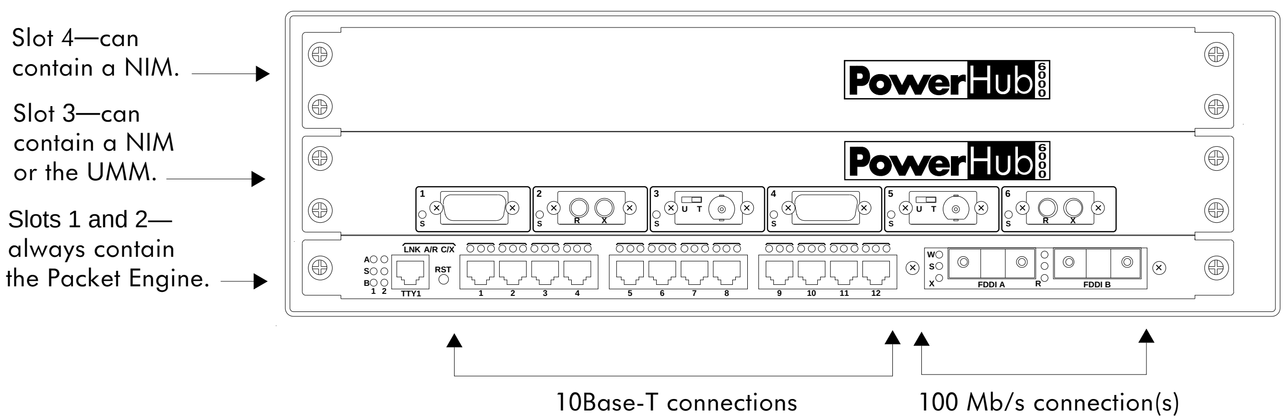


FIGURE 2–1 Front view of the PowerHub 6000.

Slots 1 and 2 are occupied by the *Packet Engine*. The *Packet Engine* is the hub's centralized packet processing and forwarding engine. It contains the "switching fabric," in which packets are examined, then dropped or forwarded as required.

The Packet Engine contains twelve 10Base-T Ethernet connections (slot 1). In addition to the twelve Ethernet connections, the Packet Engine contains an expansion slot for an optional 100 Mb/s daughter card (slot 2). The 100 Mb/s card can be a Fast Ethernet or FDDI daughter card.

Slot 3 can contain the *UMM (Universal Media Module)*, as shown in Figure 2–1. The UMM is an optional module you can install to bypass the first six UTP segments on the Packet Engine with six Ethernet segments of other interface types, such as BNC, 10Base-FB, 10Base-FL, or AUI.

Slot 4 in Figure 2–1 is empty and is covered by a cover plate. However, you can install a *NIM (Network Interface Module)* in the slot. A NIM is a plug-in module that provides additional segment connections to the hub. You also can install a NIM, rather than the UMM, in Slot 3. The UMM and NIMs are described in Section 2.5 and Section 2.4.

2.1 PACKET CHANNEL BACKPLANE

The *Packet Channel backplane* provides connection between the Packet Engine and the NIMs. Using the Packet Channel backplane, the Packet Engine and NIMs exchange packet traffic and control data. Packets forwarded to segments on a NIM are transferred along the Packet Channel backplane to the Packet Engine for processing. Likewise, packets destined for segments on a NIM are sent from the Packet Engine to the NIM along the backplane.

The Packet Channel backplane is installed inside the chassis, directly on the Packet Engine. (See Section 5.10 on page 99 for information about installing or removing the Packet Channel backplane.)

You do not need the Packet Channel backplane unless you plan to install a NIM. The UMM does not connect to the Packet Channel backplane, but instead attaches directly to the Packet Engine using ribbon cables. The segments on the Packet Engine itself do not use the Packet Channel backplane.

2.2 POWER SUPPLY

The PowerHub 6000 chassis contains one or two power supplies (depending on what you specify on your order), accessible from the rear of the chassis. Each *power supply* transforms AC current into the +5-volt and +12-volt DC current needed by the PowerHub components. The power supply runs on a standard 115-volt AC or 230-volt AC power source, and draws 5A at 115 volts and 2.5A at 230 volts. Figure 2–2 shows two power supplies installed in the PowerHub 6000 chassis.

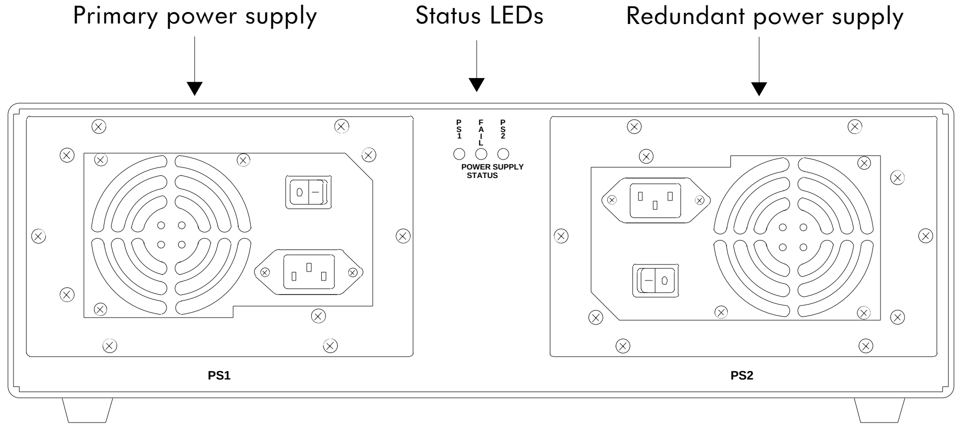


FIGURE 2–2 Rear view of PowerHub 6000 chassis, showing power supplies.

You can run any PowerHub 6000 configuration on a single power supply. However, a second power supply adds redundancy in case a power supply fails. If one of the two power supplies ever fails, the entire load is immediately assumed by the other power supply. The PowerHub 6000 continues to run with no interruption to service.

Note that the redundant power supply is upside down relative to the primary supply. This design allows the power-supply cables inside the chassis to connect to the power backplane inside the chassis.

> **NOTE**: If your chassis contains two power supplies, make sure you always turn both of them on at the same time. Moreover, if you need to power down the PowerHub 6000, make sure you switch both power supplies off.

In addition to providing redundancy, a second power supply enhances the reliability of both power supplies by load sharing. When power supplies *load share*, they participate equally in providing power to the chassis.

If a single power supply is used, it must be installed in the left power supply bay, and the protective cover plate must be installed over the unused bay. (Operating the system with an uncovered power supply bay affects internal cooling and can void the warranty.)

In the unlikely event of a short circuit or other overload condition, both power supplies automatically shut down to protect the PowerHub components from damage.

2.2.1 LEDs

Three LEDs provide status information for the power supplies. Table 2–1 lists the LEDs.

TABLE 2–1 Power supply LEDs.

Label	Color	Indicates…
PS1	Green	The primary power supply is on and functioning normally.
PS2	Green	The secondary (redundant) power supply is on and functioning normally.
FAIL	Red	One of the power supplies is not supplying current to the system. This can occur if a power supply is turned off or fails. You can identify the power supply that is turned off or has failed by observing the green status LED (PS1 or PS2). If the power supply is not supplying current, the LED is dark. If a power supply is turned off or fails, the other power supply automatically takes over with no interruption in hub operation. However, you should replace a failed power supply as soon as possible. (Make sure to turn off the other power supply first.) See Section 5.1 on page 72 for instructions on removing a failed power supply or installing a new one.

In addition to the information shown by the LEDs, status information for the power supplies and other aspects of the chassis configuration are shown by the **`mgmt showcfg`** command. (See Section 9.3.6 on page 160.)

2.3 PACKET ENGINE

The Packet Engine examines packet headers for bridging and routing, then modifies them as required for routing. When a segment connector receives a packet, the connector transfers the packet directly to the Shared Memory on the Packet Engine. Bridging and routing engines in the Packet Engine store all the packets and related data structures in the Shared Memory. This contributes to high packet throughput because all packet-related data is stored in one place. Figure 2–3 shows the front panel of the Packet Engine.

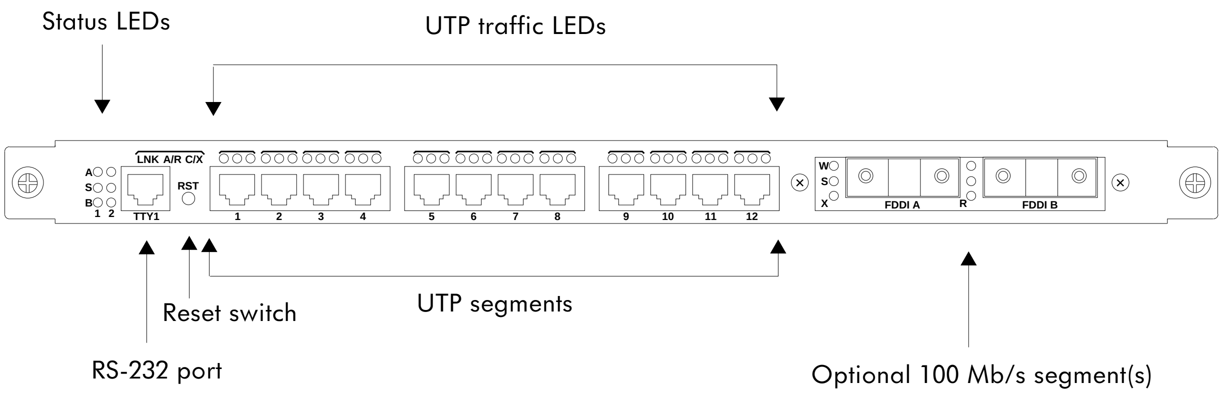


FIGURE 2–3 Packet Engine (front view).

The Packet Engine contains the following components:

- Indicator LEDs.
- RS-232 port (TTY1).
- Reset switch.
- Lock Switch and jumper. (Some Packet Engines do not have the Lock Switch.)
- Temperature sensor.
- Flash Memory Module.
- Main Memory.
- Boot PROM.
- ID PROM.
- Non-volatile RAM (NVRAM).
- Twelve Ethernet 10Base-T segments (RJ-45 connectors).
- Position for optional 100 Mb/s segment(s).

These components are described in the following sections.

2.3.1 LEDs

The Packet Engine has two sets of LEDs. The set located on the left end of the Packet Engine shows status information for the Packet Engine itself. These LEDs are described in Table 2–2. The other set show status information for the network segments attached to the 10Base-T connectors. These LEDs are described in Section 2.3.11.

TABLE 2–2 Packet Engine status LEDs.

Label	Color	Indicates…
A	Red	Alarm. These LEDs indicate that a system crash has occurred. One or both LEDs remain lit until the hub is reset by pressing the reset (RST) switch or by switching the power supplies off, then on.
S	Green	The CPUs are functioning normally. If both of these lights go out during normal operation, there might be a problem in the Packet Engine. Contact FORE Systems TAC.
B	Amber	When flashing, indicates the hub is booting. These LEDs are lit only when the system is booting, and go out as soon as the system is finished booting.

2.3.2 *RS-232 Port* (TTY1)

The Packet Engine contains an RS-232 port, labeled TTY1. You use the TTY1 port to attach a management terminal or modem to the hub. You access the user interface through the terminal or modem attached to TTY1. When you boot the PowerHub 6000, it automatically begins a command-line session on the management terminal or modem connected to the TTY1 port.

The TTY1 port uses an RJ-45 connector. This connector looks like the UTP Ethernet connectors, but cannot be used for an Ethernet connection. Do not attempt to use the TTY1 port as an Ethernet segment.

The TTY1 port fully supports the following asynchronous modem-control lines: RXD, TXD, DCD, DTR, RTS, and CTS.

You can configure the TTY1 port to support one of the following baud rates: 1200, 2400, 4800, 9600, and 19200. The default baud rate is 9600. Although you can change the baud rate of the TTY1 port, you cannot change it until you log on. Accordingly, your management terminal or modem must support 9600 baud the first time you log on.

The Packet Engine has a second RS-232 port (TTY2), which is used only by FORE Systems TAC for diagnosing system problems. The TTY2 port is not accessible from the Packet Engine front panel. Do not attempt to use the TTY2 port unless instructed to do so by FORE Systems TAC.

To ensure session security, both TTY ports use DCD (data carrier detect). If the terminal or modem connected to a TTY port is supplying the DCD signal, the hub monitors that signal. If DCD is dropped, the software logs you out of the user session, preventing other terminals or modems from connecting to the session.

If the terminal or modem attached to the hub does not supply a DCD signal, you still can use the TTY port. The hub emulates the DCD signal internally. (However, if the cable contains another signal where the hub expects to find the DCD signal, the cable might prevent proper operation of the TTY port.)

See Section 4.3.2 on page 55 for instructions on attaching your management terminal or modem to the TTY1 port. This section describes how to assemble an RS-232 cable to connect to the TTY1 port. If you already have an RS-232 cable, you can use the pinout information in this section to verify that your cable is compatible with the hub.

2.3.3 *Reset Switch* (RST)

The reset switch is located to the right of the TTY1 port and is labeled RST. When you press the reset switch, the Packet Engine performs a “cold” restart of the hub. During a cold restart, the Packet Engine conducts a power-on self-test to check its various hardware components.

Depending on the boot preference(s) you specify, the Packet Engine uses files in the Flash Memory Module or on a TFTP server (network booting) to configure the hub for runtime operation. When shipped from the factory, the PowerHub software is configured to boot from the Flash Memory Module. If your chassis does not contain the Flash Memory Module, you will need to configure the hub for netbooting.

To change the boot source, use the instructions in Section 4.3.4 on page 60.

2.3.4 Lock Switch and Jumper

The Lock Switch lets you lock the hub so that a login ID and password are required before a user can access the hub's command-line interface. The switch is set to Unlocked at the factory, which means that you or anyone else who is capable of establishing an RS-232 or TELNET connection to the hub can access the user interface. The Lock Switch is located on the front panel, to the left of the reset (RST) button.

Some chassis do not contain an external Lock Switch. However, all chassis contain a Lock Switch jumper on the Packet Engine. You can lock or unlock the hub by changing the setting of the Lock Switch jumper. This jumper is set to Unlocked at the factory but you can set it to Locked.

When the Lock Switch (or Lock Switch jumper) is set to Locked, users must enter a management capability at the `login:` prompt and the password for that capability before they can begin the session and issue commands. The *management capability* can be *monitor* or *`root`*:

- *Monitor* lets you display information, but does not allow statistics to be cleared or configuration settings to be changed.
- *Root* lets you issue any command, including commands that clear statistics and change the PowerHub configuration.

See Section 5.12 on page 103 for information about moving the Lock Switch jumper. See Section 8.4 on page 139 for information about setting a password.

2.3.5 Temperature Sensor

The Packet Engine contains a temperature sensor that reads the temperature inside the chassis within an accuracy of plus or minus 0.5˚ C. You can use the **`mgmt temperature`** command to display the current temperature of the hub. (See Section 9.4.2 on page 164.)

2.3.6 Flash Memory Module

The *Flash Memory Module* is an optional EEPROM device used by the hub for file storage. The boot PROM and system software contain code that implements a DOS-like file system on the module.

You can use the file system to display, copy, rename, and remove files in the module. In addition, you can display directory information for files. However, the file system does not support hierarchical directory structures.

The Flash Memory Module comes in two sizes: 2 MB and 4 MB.

All of the files required to boot and run the PowerHub 6000 software can be stored on the Flash Memory Module. If you do not plan to boot the hub over the network (netbooting), the hub must contain the module.

Files stored on the Flash Memory Module can be read faster than files read over the network. Thus, even in networks that support netbooting, booting from the Flash Memory Module is faster.

To order the Flash Memory Module, contact FORE Systems TAC or your reseller. For instructions about installing the Flash Memory Module, see Section 5.5.1 on page 81. For information about managing files on the Flash Memory Module, see Section 9.6 on page 176.

2.3.7 Main Memory

The Packet Engine comes with 8 MB of main memory (DRAM). The main memory is used by the software's bridging and routing engines. With the standard 8 MB of memory, you can run bridging, IP routing, and any two of the other supplied protocols or the Bridge MIB, simultaneously. An upgrade to 16 MB of memory enables you to run all supplied protocols and the Bridge MIB simultaneously.

Table 2–3 details the protocols you can use simultaneously with 8 MB or 16 MB of main memory.

TABLE 2–3 Protocol Support.

Total DRAM	Protocols Supported Simultaneously
8 MB	Bridging and IP routing, plus any two of the following: IPX, AppleTalk, DECnet, Bridge MIB
16 MB	Bridging and IP routing, plus all four of the following: IPX, AppleTalk, DECnet, Bridge MIB, OSPF This includes systems containing FDDI daughter cards.

For information on installing a DRAM upgrade, see Section 5.7.

2.3.8 Boot PROM

The Packet Engine has a *boot PROM* that contains software used by the PowerHub 6000 when it is booted. You issue software commands to the boot PROM to perform configuration tasks such as specifying the boot source and installing software upgrades. The command prompt for the boot PROM looks like this: `<PROM-6pe>`.

See Chapter 10 for a description of the boot PROM commands.

2.3.9 ID PROM

The Packet Engine and all types of NIMs have an *ID PROM* that contains identification information for the module, and lists the maximum amount of current required to power the module. The ID PROM lists the module's serial number, model number, Rev (revision) number, and other factory-issued information. You can display the contents of the ID PROM using the **`mgmt idprom`** command (see Section 9.4.1).

2.3.10 NVRAM

The *NVRAM* (Non-Volatile RAM) maintains the system time and date and contains configuration information such as the baud rate of the TTY1 port, the PowerHub system name, and the boot source (Flash Memory Module or TFTP file server).

You can access NVRAM from either the boot PROM's command prompt (<PROM-6pe>) or the run-time command prompt. See Chapter 11 for information about the commands you can use to configure the NVRAM.

2.3.11 10Base-T Connectors

As shown in Figure 2–3, the Packet Engine contains twelve RJ-45 UTP connectors. These connectors provide the mechanical connection, pinouts, and electrical functions of a standard IEEE 802.3 10Base-T hub.

Three LEDs above each connector give connection status and traffic information for the network segment attached to the connector. The information they display depends on their configuration. Table 2–4 describes the information shown by the UTP LEDs.

TABLE 2–4 UTP LEDs, on Packet Engine.

Label	Color	Indicates…
C/X	Amber	Transmit collisions or packet transmissions, depending upon the setting. Receive collisions (collisions occurring while a segment is not transmitting) do not occur on private UTP segments such as those on the Packet Engine. • When set to C (transmit-collision mode), this LED indicates transmit collisions. Each time a transmit collision occurs on the corresponding segment, this LED is illuminated for 5 – 10 ms. • When set to X (transmit mode), this LED indicates packet transmission. Each time a packet is transmitted on the corresponding segment, this LED is illuminated for 5 – 10 ms.
A/R	Green	Packet activity (transmit and receive) or receive activity only, depending upon the setting: • When set to A (activity mode), this LED indicates when a packet is transmitted or received. Each time a packet is transmitted or received on the corresponding segment, the LED is illuminated for 5 – 10 ms. • When set to R (receive mode), this LED indicates when packets are received. Each time a packet is received on the corresponding segment, the LED is illuminated for 5 – 10 ms.
LNK	Green	Link status. When a LNK LED is glowing, the corresponding segment is successfully detecting link-test pulses through the twisted-pair wire attached to the segment. A LNK LED goes dark if you remove the twisted-pair cable attached to the corresponding segment. If the cable is attached but the LED is still dark, a problem might exist in the segment or in another device attached to the cable.

The traffic LEDs (C/X and A/R) are configured as a pair. You can configure them as C and A or as X and R. The default setting is C and A. You can configure them as a group, but not individually on a segment-by-segment basis. The configuration of the LNK LED cannot be changed.

The optional Fast Ethernet daughter cards (described in Section 2.3.12) also use these LEDs. When you configure the LEDs, the configuration applies to all of the UTP and Fast Ethernet segments in slots 1 and 2 (the Packet Engine and the daughter card).

2.3.12 100 Mb/s Daughter Card

In addition to the twelve UTP segments, you can add a 100 Mb/s Fast Ethernet or FDDI daughter card to the Packet Engine. The opening for the card is located on the right side of the Packet Engine. (See Section 5.3.1 on page 77 for instructions about installing a 100 Mb/s daughter card.)

2.3.12.1 Fast Ethernet

The Fast Ethernet daughter cards provide connection to high-performance workstations or to upstream hubs. The PowerHub 6000 supports three different 100Base standards for different physical media types:

100Base-TX Uses two pairs of CAT-5 (data-grade) wiring, one for transmit and one for receive. Because there are separate pairs for transmit and receive, full-duplex operation is possible. Connection is provided by an RJ-45 connector.

100Base-FX For multimode, provides connection to two multimode fiber-optic cables with ST connectors. For single-mode, provides connection to two single-mode fiber cables with an SC connector. In each case, one cable is used for transmit and the other is used for receive, so full-duplex operation is possible.

100Base-T4 This uses four pairs of CAT-3 (voice-grade) or better wiring. All four pairs are used while sending packets in one direction, so full-duplex operation is not possible.

The pinouts in the 100Base connectors differ according to how the connector is used:

- If you connect to a workstation, use a straight cable.
- If you connect to an upstream hub, use a cross-over cable.

You can install a daughter card containing a single Fast Ethernet segment, two Fast Ethernet segments of the same type, or a combination of one 100Base-TX segment and one 100Base-FX segment. Additional combinations might become available in the future.

Figure 2–4 shows the front panel of the dual 100Base-TX Fast Ethernet daughter card.

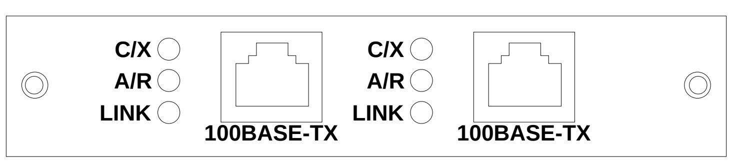


FIGURE 2–4 Dual 100Base-TX Fast Ethernet daughter card (front view).

Figure 2–5 shows the front view of the dual 100Base-FX Fast Ethernet daughter card.

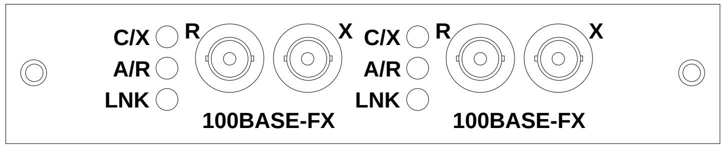


FIGURE 2–5 Dual 100Base-FX Fast Ethernet daughter card (front view).

As shown in these figures, the link status and traffic LEDs for the 100Base connectors show the same information as the LEDs for the 10Base-T connectors (see Section 2.3.11). Note that all the traffic LEDs for the 100Base and 10Base-T connections in the same hub use the same configuration, either C/X or A/R. You cannot configure them individually.

If your configuration requires a combination of 100Base-TX and 100Base-FX connections, you can install a mixed Fast Ethernet card, such as the one shown in Figure 2–6.

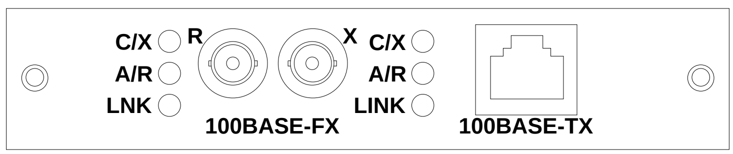


FIGURE 2–6 Mixed 100Base-FX and 100Base-TX Fast Ethernet daughter card (front view).

2.3.12.2 FDDI

The PowerHub 6000 supports the ANSI X3T9.5 FDDI standard. The following types of FDDI daughter cards are available for the PowerHub 6000:

- Single DAS.
- Single SAS.

Each of these cards provides a single FDDI segment. Depending on the card type, these daughter cards provide connection to one or two FDDI ports. You can use these ports for any of the following types of FDDI connections:

- A DAS attachment to your FDDI backbone. In this configuration, the ports on the FDDI daughter card function as A and B ports.
- A null-attached concentrator. In this configuration, both ports are used as M ports to attach to two SAS stations.
- A dual-homed device. Both ports are attached to the M ports of two separate concentrators in the same ring. In this configuration, the FDDI ports are each configured as S ports. This configuration provides backup in case one of the connections fails. The B port (FDDI B) has precedence, but if the B port or the concentrator it is attached to fails, the A port (FDDI A) takes over automatically.

Figure 2–7 shows the front panel of the Single DAS FDDI daughter card. The card in this example contains MICs, used for connection to multimode fiber.

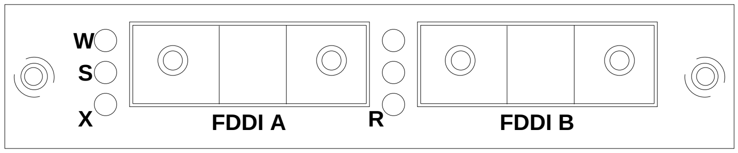


FIGURE 2–7 Multi-mode Single DAS FDDI daughter card (front view).

If your cable plant uses single-mode fiber, you can use the single-mode FDDI daughter card, which provides ST connectors. The single-mode FDDI daughter card provides the same features as the multi-mode FDDI daughter card, but attaches to your single-mode fiber-optic cable.

Each FDDI port has three connection LEDs. The LEDs show the same information for single-mode and multimode connections. Table 2–5 list the FDDI connection LEDs.

TABLE 2–5 FDDI connection LEDs.

Label	Color	Indicates…
W	Amber	Wrap. For a DAS connection, its left or right "Wrap" LED is lit when the connection is in the wrap-A or wrap-B state respectively. If only one A or B port is being used as a SAS connection, its W LED *and* S LEDs are lit.
S	Green	Status. For a normal DAS connection (neither port is wrapping), both S LEDs are lit. If only one port is being used (as a SAS connection) its W LED is lit as well.
X	Green	Transmit. Associated with the entire FDDI segment, regardless of whether one or both ports are in use. The X LED indicates when the hub is transmitting packets onto the attached segment.
R	Green	Receive. Like the X LED, this LED is associated with the entire FDDI segment, regardless of whether one or both ports are in use. The R LED indicates when the hub is receiving packets from the attached segment.

2.4 NETWORK INTERFACE MODULES

NIMs (Network Interface Modules) are optional modules you can install in the PowerHub chassis to increase the number of segment connections on the hub. You can install NIMs in slots 3 and 4. An installed NIM occupies the entire slot. The following types of NIMs are available for the PowerHub 6000:

- 12x1 10Base-T Ethernet (using RJ-45 or Champ connectors).
- 24x1 10Base-T Ethernet (using RJ-45 or Champ connectors).
- 6x1 10Base-FL Ethernet.
- 12x1 10Base-FL Ethernet.

Note that you must install the Packet Channel backplane to use the NIMs. (See Section 2.1.) The NIMs are described in the following sections.

2.4.1 10Base-T NIMs

The PowerHub 6000 supports the following types of 10Base-T NIMs:

- 12x1 10Base-T RJ-45 Ethernet.
- 24x1 10Base-T RJ-45 Ethernet.
- 12x1 10Base-T Champ Ethernet.
- 24x1 10Base-T Champ Ethernet.

2.4.1.1 10Base-T RJ-45 NIMs

The 12x1 10Base-T RJ-45 NIM adds connection for 12 10Base-T Ethernet segments to the hub. The function and features of the segments in the NIMs are identical to those of the 10Base-T segments on the Packet Engine.

The 24x1 10Base-T RJ-45 NIM uses the same connection and LEDs as the 12x1 but provides connection to 24 10Base-T segments.

Figure 2–8 shows the front view of the 12x1 10Base-T RJ-45 NIM.

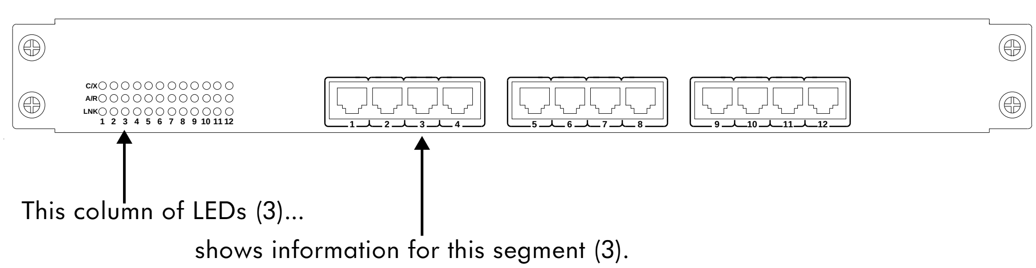


FIGURE 2–8 12x1 10Base-T RJ-45 NIM (front view).

Each RJ-45 connector on the NIM provides connection to an independent 10 Mb/s 10Base-T segment. You can place individual segments into full-duplex mode using the **`mgmt operating-mode`** command. (See Section 9.5.6 on page 175.)

The segment LEDs are organized into columns on the left side of the NIM. Each column shows link status and traffic information for one segment. The number underneath an LED column indicates the segment to which the LEDs correspond.

The connector numbers on the NIM faceplates merely associate each connector with its LEDs. The numbers do not correspond to the segment numbers recognized by the software. For example, network segment 7 is always associated with UTP connector 7 on the Packet Engine, not with connector number 7 on a NIM. The segment numbers depend on the configuration of the PowerHub 6000 chassis. (See Section 1.3 on page 17.)

2.4.1.2 12x1 and 24x1 Champ Modules

The Champ modules provide 10Base-T Ethernet connections. The 12x1 has a single Champ connector with 12 independent 10Base-T segments. The 24x1 has two Champ connectors. Each Champ connector provides connection to 12 independent 10Base-T segments for a total of 24 segments.

Figure 2–9 shows the 12x1 Champ NIM.

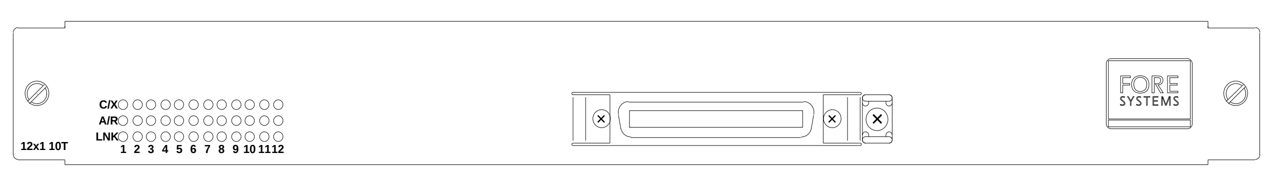


FIGURE 2–9 12x1 Champ NIM (front view).

Figure 2–10 shows the 24x1 Champ NIM.

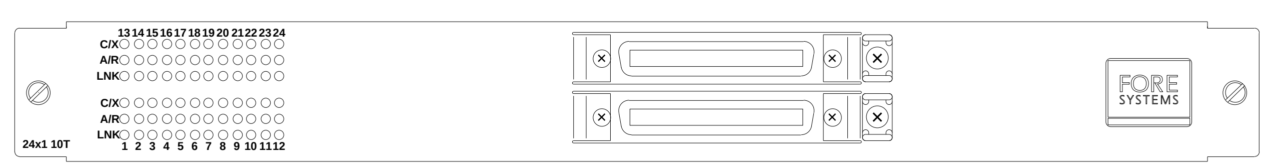


FIGURE 2–10 24x1 Champ NIM (front view).

Figure 2-11 shows the pin locations on each Champ connector. Note that the leftmost pins, 25 and 50, are not connected to a UTP segment; they are unused.

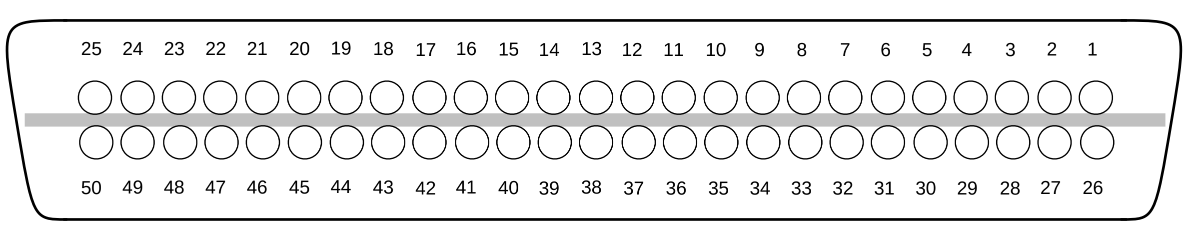


FIGURE 2-11 Pin locations on Champ connector.

Figure 2-12 shows the location of the pin signals for each segment on the Champ connector. Use this information along with the pinouts listed in Section C.4 on page 249 to wire your network cables for the Champ segments. The segment numbers are for the Champ connector on the 12x1 NIM and the bottom Champ connector on the 24x1 NIM. The segment numbers in parentheses are for the top Champ on the 24x1.

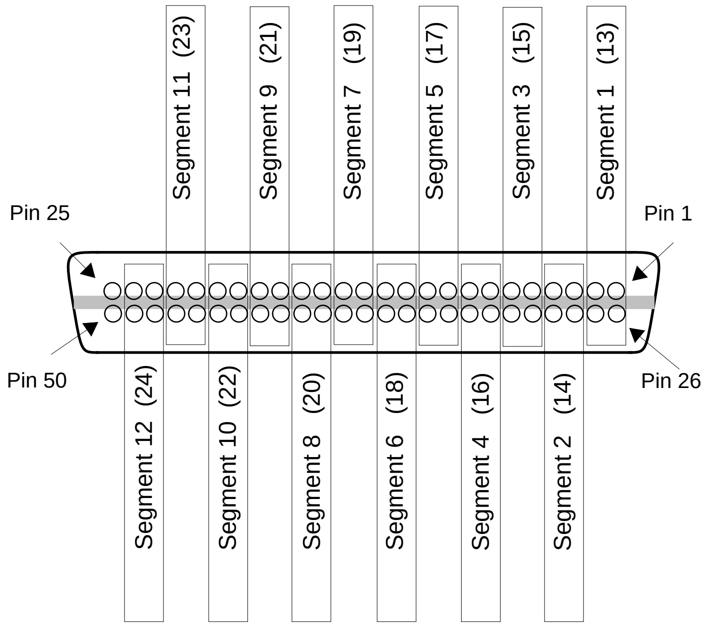


FIGURE 2-12 Segment positions on Champ connector.

2.4.2 10Base-FL NIMs

The 10Base-FL NIMs add 10Base-FL Ethernet segments to the PowerHub 6000 chassis. These segments provide 10Base-FL connections that are FOIRL (fiber optic inter-repeater link) compatible. The signal is compatible with legacy equipment that uses the FOIRL standard. For each segment, connection is provided by a pair of ST Fiber connectors. The connector on the left side receives and the connector on the right side transmits.

The 6x1 10Base-FL NIM adds six 10Base-FL Ethernet segments and the 12x1 10Base-FL NIM adds 12 10Base-FL Ethernet segments. Figure 2–13 and Figure 2–14 show the 6x1 10Base-FL NIM and the 12x1 10Base-FL NIM.

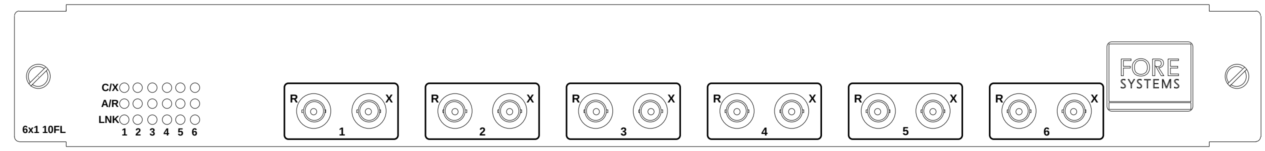


FIGURE 2–13 6x1 10Base-FL NIM (front view).

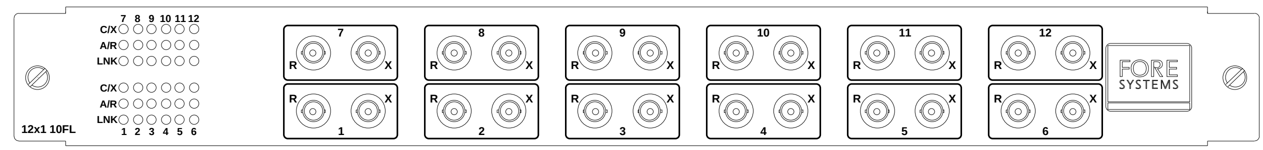


FIGURE 2–14 12x1 10Base-FL NIM (front view).

2.5 UNIVERSAL MEDIA MODULE

The UMM (Universal Media Module) lets you bypass the first six UTP segments on the Packet Engine and connect to different types of Ethernet segments in their place. You can install one UMM in the PowerHub 6000 chassis. The UMM must be installed in the slot directly above the Packet Engine. Figure 2–15 shows an example of a PowerHub 6000 chassis containing the UMM.

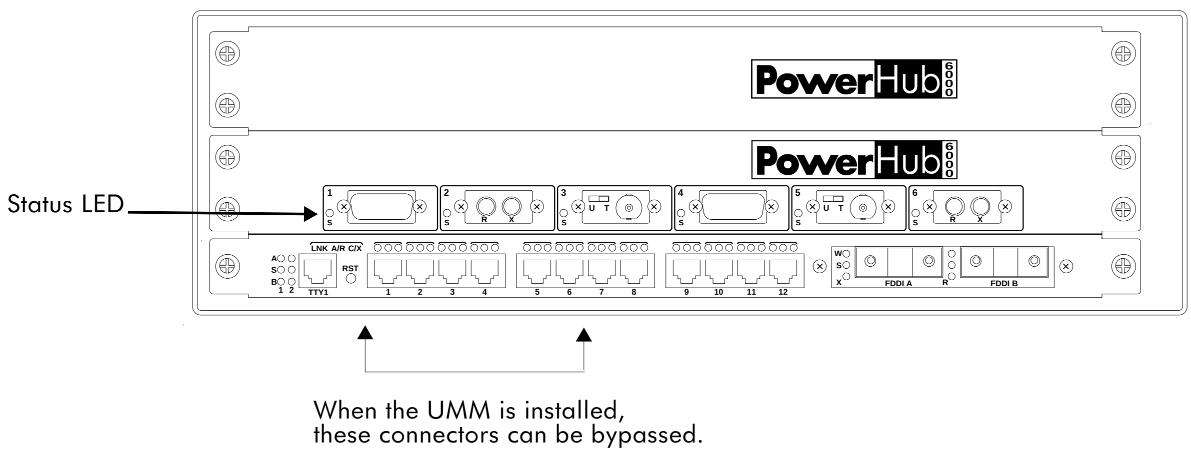


FIGURE 2–15 Segments 1 through 6 are bypassed by UMM segments.

You can configure the UMM with any combination of the following Ethernet media types:

- 10Base-FL (FOIRL-compatible).
- 10Base-FB.
- AUI (10Base5 or any other external transceiver type).
- BNC (10Base2).

The 10Base-FB, 10Base-FL, and BNC connections are provided by *EMAs (Ethernet Media Adapters)*, small daughter cards that you install directly onto the UMM. The AUI connection is provided by an AUI Media Cable, which also attaches directly to the UMM.

NOTE: The PowerHub 6000 does not support 10Base-T EMAs or MAU EMAs.

You do not need the Packet Channel backplane to use the UMM. The daughter cards on the UMM are connected directly to the Ethernet electronics on the Packet Engine by two ribbon cables. (Section 5.8.1 on page 90 describes how to install the UMM and EMAs.)

When the UMM is installed and active, the first six UTP connectors on the Packet Engine, counting from left to right, are bypassed. Traffic that normally would flow to and from these connectors instead flows to and from the corresponding EMAs or AUI Media Cables on the UMM.

When you boot the hub, the software senses whether you have installed the UMM and automatically activates the EMAs or AUI Media Cables in the UMM, and disables the corresponding UTP segments on the Packet Engine.

The LEDs next to each bypassed UTP connector provide collision and traffic information for the corresponding EMA or AUI Media Cable. For example, the LEDs for UTP connector 1 provide information for the EMA in position 1 (far left) on the UMM. In addition to these LEDS, a green Status (S) LED in the lower left corner of each connection on the UMM glows when the EMA has been activated. (You can override an EMA or AUI Media Cable and re-activate the UTP connector using the **`mgmt activate-utp`** command. See Section 9.5.2 on page 166.)

The following sections describe the AUI Media Cable and the EMAs.

2.5.1 AUI Media Cable

The *AUI Media Cable* is a cable that provides a DB-15 connector on the UEM. This DB-15 connector provides the mechanical connection, pinouts, and electrical functions of a standard IEEE 802.3 AUI connection. The standard AUI connection, in turn, connects to a 10Base5 "thick net" or other transceiver (MAU).

Figure 2–16 shows an AUI Media Cable.

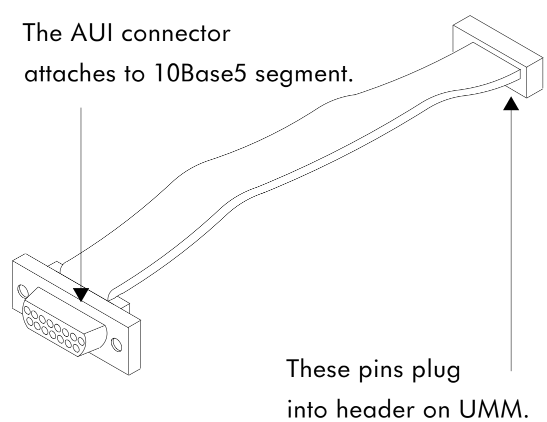


FIGURE 2–16 AUI Media Cable.

The pin connector at the end of the cable (shown on the right side of Figure 2–16) connects to a header on the UMM. The connector has slightly fewer pins than an EMA connector and plugs into the header behind the EMA header on the UMM. The AUI connector attaches to a MAU (Medium Access Unit) transceiver or a standard AUI straight-through cable.

Note that FORE Systems' AUI Media Cable is different from a standard DB-15 flat cable commonly purchased from electronics suppliers. The PowerHub 6000 cannot detect the AUI connection if a "standard" cable is installed.

A green S (Status) LED, located on the UMM front panel to the lower left of the AUI Media Cable, indicates that the AUI Media Cable has been activated. When the AUI Media Cable has been activated, the LNK (Link) LED for the UTP segment bypassed by the Cable shows the link status for the cable. The PowerHub 6000 uses "loss of carrier" to determine the AUI diagnostic state. If most transmissions result in loss-of-carrier errors, then the segment state is determined to be "bad." When the segment state is bad, the LNK LED is dark. Note that the S LED on the UMM is not affected by the segment state.

2.5.2 10Base-FL EMA

The *10Base-FL EMA* provides a single 10Base-FL connection that is FOIRL (fiber optic inter-repeater link) compatible. This EMA provides a signal compatible with legacy equipment that uses the FOIRL standard. Figure 2–17 shows a 10Base-FL EMA.

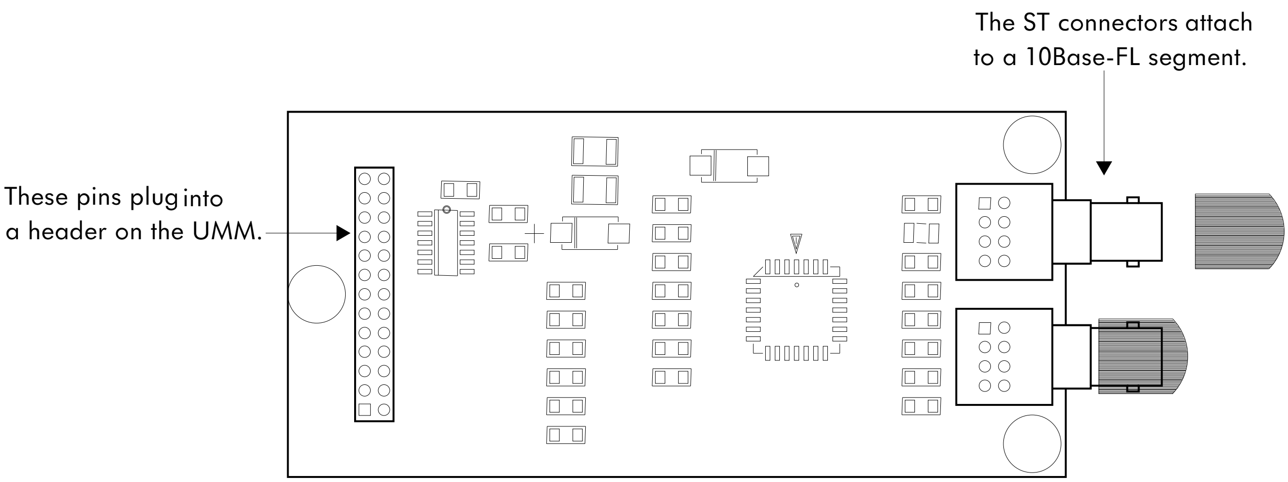


FIGURE 2–17 10Base-FL EMA (actual size).

The rows of pins shown on the left side of the card plug into a header on the UMM. The card is installed component-side down. The two ST Fiber connectors shown on the right side of Figure 2–17 attach to your 10Base-FL cables. The connector shown on the top is gray-colored and is the receive connector. When installed on the UMM, it is on the left and labeled R. The connector shown on the bottom is the transmit connector and is colored white. When installed on the UMM, it is on the right and labeled X. Make sure you attach your 10Base-FL cables accordingly.

If you need to know the optical power budget for the cabling used with the 10Base-FL EMAs, use the power values listed in Table 2–6 as a guideline.

TABLE 2–6 10Base-FL power values.

Cable type	Minimum and maximum power
62.5/125	Minimum average transmit power: –20 dBm. Maximum average transmit power: –12 dBm. Minimum average receiving power: –32.5 dBm.
50/125	Minimum average transmit power: –23.8 dBm. Maximum average transmit power: –15.8 dBm. Minimum average receiving power: –32.5 dBm.

As shown in this table, the minimum average receiving power of the 10Base-FL EMA is the same regardless of which type of cable you use. However, the minimum and maximum average transmit power differs depending upon the cable type. If you use 62.5/125 cable, the transmitted power is approximately 3.8 decibels lower than if you use 50/125 cable. Consequently, you have a smaller optical budget if you use 50/125 cable. Note that a 10Base-FL segment can span up to 2 km, but the actual effective distance might be less depending upon the number of splices in the cable.

When shipped, this card has rubber caps over the connectors to protect them from dust. Do not remove these caps until you are ready to attach your network segments to the connectors. Always replace the caps when no segments are attached.

The 10Base-FL EMA supports full-duplex operation, which you can enable using the **`mgmt operating-mode`** command (see Section 9.5.6 on page 175).

A green S (Status) LED, located on the UMM front panel to the lower left of the EMA, indicates that the EMA has been activated. When the 10Base-FL EMA is activated, the LNK (Link) LED for the UTP segment bypassed by the EMA shows the link status for the EMA. The PowerHub 6000 uses standard 10Base-FL link-beat pulses to determine the state of the 10Base-FL segment.

As in the case of the AUI Media Cable, the S LED on the UMM is not affected by the segment state.

2.5.3 10Base-FB EMA

The *10Base-FB EMA* provides a single 10Base-FB connection. In accordance with the 10Base-FB standard, the 10Base-FB EMA increases the number of repeaters allowed between end stations from five (as in 10Base-FL) to 12.

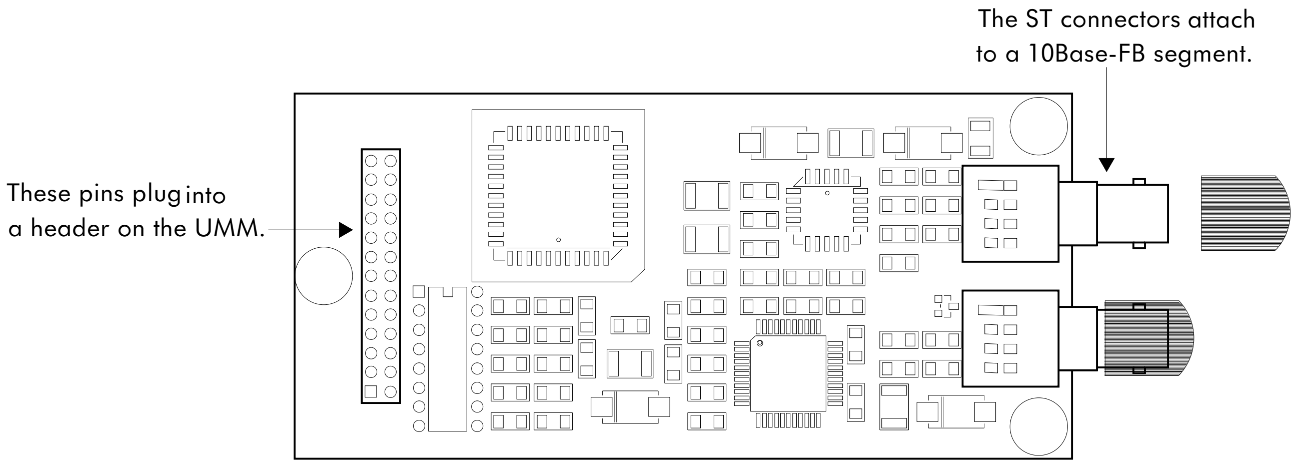


FIGURE 2–18 10Base-FB EMA (actual size).

As shown in this figure, the 10Base-FB EMA, like the 10Base-FL EMA, uses two ST Fiber connectors. These connectors attach to your fiber optic cables. As in the case of the 10Base-FL EMA, the connector shown on the top is the receive connector. When installed on the UMM, it is on the left and labeled R. The connector shown on the bottom is the transmit connector. When installed on the UMM, this connector is on the right and labeled X. Make sure you attach your 10Base-FB cables accordingly. The power values for the 10Base-FB are the same as those for the 10Base-FL. If you need to install cabling for the 10Base-FB, see Table 2–6.

When shipped, the 10Base-FB EMA card has rubber caps over the connectors to protect them from dust. Do not remove these caps until you are ready to attach your network segments to the connectors. Always replace the caps when no segments are attached.

A green S (Status) LED indicates that the EMA has been activated. When the 10Base-FB EMA is activated, the LNK (Link) LED for the UTP segment bypassed by the EMA shows the link status for the EMA. As in the case of the 10Base-FL EMA, the S LED on the UMM is not affected by the segment state.

2.5.4 BNC EMA

The *BNC EMA* provides a single 10Base2 “thin net” coax connection. Each BNC EMA contains an internal 50-ohm termination resistor you can activate using the switch on the front panel of the EMA. The switch has two positions: `U` and `T`. `U` means “unterminated”; `T` means “terminated.” The position of the switch also can be sensed by the software and displayed using the **`mgmt showcfg`** command. (See Section 9.3.6 on page 160.) Figure 2–19 shows a BNC EMA.

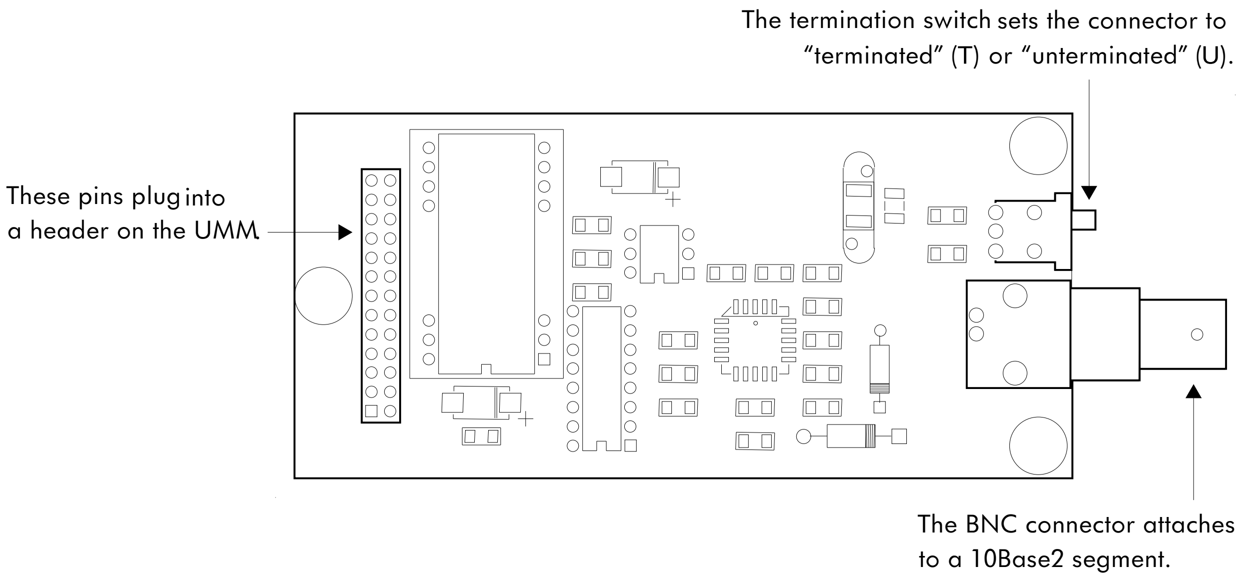


FIGURE 2–19 BNC EMA (actual size).

The rows of pins shown on the left side of Figure 2–19 plug into a header on the UMM. You attach a coaxial cable to the BNC connector shown on the right side of Figure 2–19. The BNC terminator switch is located to the left of the BNC connector.

A green `S` (Status) LED, located on the UMM front panel to the lower left of the EMA, indicates that the EMA has been activated. When the BNC EMA is activated, the `LNK` (Link) LED for the UTP segment bypassed by the EMA shows the link status for the EMA. The PowerHub 6000 uses heuristic methods to determine the BNC diagnostic state. If no packets are received from the segment, and most transmissions sent on the segment result in collisions, then the segment state is determined to be bad.

The `S` LED on the UMM is not affected by the segment state.

3 Software

This chapter describes the different types of software used by the PowerHub 6000, and describes the files on the Flash Memory Module and system software diskettes shipped with the hub. The software is already installed on the Flash Memory Module. If your system does not contain the Flash Memory Module, you will need to install the software using to the instructions in Section 4.3.4 on page 60.

3.1 TYPES

The PowerHub modules use the following types of software:

Packet Engine boot PROM

Used by the Packet Engine when it is booted. You can access the Packet Engine boot PROM and change configuration values—including the boot source—stored in NVRAM. The command prompt for the Packet Engine boot PROM looks like this:

```
<PROM-6pe>
```

System software

Sometimes called “runtime software.” Contains most of the commands and features documented in this manual. You access the runtime software from the runtime command prompt. Here is an example of the prompt:

```
1:PowerHub:
```

For information about the command prompts and the user interface, see Section 7.3 on page 125.

3.2 FILES

The system software is shipped on the Flash Memory Module (if your system was configured with one at the factory) and on two 3-1/2” 1.4 MB HD floppy diskettes. The diskettes are identical.

- If the hub is shipped with the Flash Memory Module, the system software is already installed on the module at the factory. The diskettes contain your backup copies.
- If the hub is not shipped with the Flash Memory Module, the system software resides only on the floppy diskettes. Before you can use the hub, you must load the system software onto a file server and configure the hub for network booting, as described in Section 4.3.4 on page 60. Alternatively, you can order and install the Flash Memory Module.

> **NOTE**: If you need to install PowerHub files onto a TFTP server or the Flash Memory Module, see Chapter 6.

The following files are provided:

`6pe`	The latest version of the system software “image” file. When you boot the hub, this file is loaded into the Packet Engine’s main memory (DRAM).
`intloop`	When you issue the appropriate command, this script performs a loopback test of the internal hardware of the segment connectors.
`extloop`	When you issue the appropriate command, this script performs an external loopback test of the send and receive hardware in the segment connectors. Whereas the `intloop` script tests only the internal hardware of the segments, the `extloop` script also tests the external send and receive hardware.
`bootdef`	When you boot the hub, this file is used to identify the system software image file (ex: `6pe`). If you boot from a file server, this file also identifies the directory in which the image file is stored.
`bootdef.fdi`	Similar to the `bootdef` file, except the `bootdef.fdi` file also loads the FDDI software.
`ppu-6pe`	The PROM Programming Utility, which you use to install upgrades to the Packet Engine boot PROM.
`bootdef.ppu`	This file causes the PowerHub 6000 to boot the *PROM Programming Utility* instead of the system software. You use the PROM Programming Utility to install upgrades to the Packet Engine boot PROM.

`dispcfg` — A configuration file that displays configuration information and statistics on the management terminal when you read the file (using the **`mgmt readcfg dispcfg`** command). If you have problems with your PowerHub system, FORE Systems' TAC might ask you to read the `dispcfg` file and send them the results. See Section 9.8.6 on page 191 for an example of the `dispcfg` file.

In addition to the files shipped with the system, the following types of files can be created and saved onto the Flash Memory Module or a file server during hub operation:

`cfg` — The default configuration file. Configuration files contain PowerHub commands that configure the hub. You create or change the default configuration file by issuing the **`mgmt savecfg cfg`** command. When you reboot the hub, the software looks for the `cfg` file. The software reads this file (if found) and executes the PowerHub commands contained in the file, configuring the hub the way it was configured when you saved the `cfg` file. See Section 9.8.1 on page 184 for an example of a configuration file.

`root.env` — The default environment file for root capability. (See Section 2.3.4 on page 27.) Environment files contain PowerHub commands that define command aliases, settings for scroll control, and timed commands. These settings affect the user session to which they are applied, but do not affect other user sessions. The `root.env` file is created when you issue the **`main saveenv root.env`** command from within a command-line session under root capability. The file is read when you begin a user session under root capability.

`monitor.env` — Similar to the `root.env` file, except this file contains the default environment settings for sessions under monitor capability. To create the `monitor.env` file, issue the following command: **`main saveenv monitor.env`**.

`fore.dmp` — If the hub contains the Flash Memory Module, this file is created automatically if the hub crashes for any reason. This file contains dump data that can be interpreted by FORE Systems TAC. If you report problems with the hub to FORE Systems TAC, they may ask you for a copy of this file along with your `cfg` file and `dispcfg` file.

Note that if the Flash Memory Module contains an `fore.dmp` file, and the hub crashes, a new `fore.dmp` file is created and the new file replaces the older one.

You can tell that the hub has crashed because one or both of the red Alarm LEDs (A) on the Packet Engine are lit. If you notice that an Alarm LED is lit, check the Flash Memory Module for an `fore.dmp` file.

If the hub does not contain the Flash Memory Module, dump files are not created.

3.3 UPGRADES

Occasionally, you might need to install an upgrade of one or more of the types of software listed in Section 3.1. The latest generally released versions of the software are installed on the PowerHub modules at the factory and provided on the software diskettes. However, if you need to install an upgrade, the upgrade is packaged on a floppy diskette.

Software upgrades are shipped in upgrade kits. These kits include Maintenance Notes or Upgrade Notes and software diskettes containing image files for each type of software being upgraded. In addition, for upgrades of the Packet Engine boot PROM, the diskette might also contain an upgrade of the PROM Programming Utility.

See Chapter 6 for instructions on installing a software upgrade.

Part 2: Installation

This part contains procedures for installing your PowerHub 6000 chassis and booting the software. This part also contains procedures for changing or adding modules in the chassis, changing the setting of the Lock Switch jumper, and installing software upgrades.

This part contains the following chapters:

Chapter 4: Installing the Hub

Contains complete procedures for installing the PowerHub 6000 and booting the software.

Chapter 5: Changing the Hardware

Contains procedures for changing or adding modules and add-on cards, changing the setting of the Lock Switch jumper, and adding DRAM to the Packet Engine.

Chapter 6: Installing Software Upgrades

Contains procedures for installing an upgrade for the Packet Engine boot PROM or the system software.

4 Installing the Hub

This chapter explains how to install the PowerHub 6000 chassis and how to boot the software. You need to complete the procedures in this chapter before you can use the hub.

- For information on re-configuring the hardware in an installed chassis, including removing individual modules or installing optional modules, see Chapter 5.
- For information on installing software upgrades, see Chapter 6.

> **CAUTIONS**: Electrostatic discharges (shocks) can permanently damage PowerHub components. We recommend that you wear an ESD (electrostatic discharge) strap when performing the hardware procedures in this chapter. Carefully read all of Section 4.1 on page 49 for details.
>
> Some procedures require the use of a #2 Phillips-head screwdriver. Do not use a #1 Phillips-head screwdriver for these procedures; it can strip the screw heads. A #2 screwdriver is larger than a #1 screwdriver and can be identified by its tip, which is flat rather than pointed.

4.1 SAFETY AND HANDLING PRECAUTIONS

Use care and common sense when handling PowerHub modules. Improper handling of PowerHub modules can result in damage to their components or injury to yourself.

To avoid personal injury:

- Do not immerse components in water or any other liquid.
- Do not stand on a wet surface while inserting or removing PowerHub modules.
- Always cover unused slots or bays with the supplied cover plates. Never place tools, your hand, or any other body part inside empty power supply bays or module slots.

4.1.1 Electrostatic Discharge

All electronic components, including PowerHub components, can be damaged by improper handling. One of the most common, although unintentional, types of mishandling is *electrostatic discharge* (*ESD*).

Electrostatic discharge can occur when you and the equipment you are handling are at different voltage potentials. When you come into contact with the equipment, the difference in your potentials causes energy to be passed from you to the component, delivering a shock to the component.

The human body is a good conductor of electricity and can deliver shocks containing thousands of volts. In fact, most people perceive a static shock only when the voltage of the shock is at least 6,000 volts. However, many electronic components can be damaged by shocks as low as 2,000 volts!

4.1.2 Guarding Against Damage

To guard against damaging PowerHub modules, always take the following precautions when handling the modules:

- Wear an anti-static wrist guard. To use the wrist guard, securely fasten the strap end to your wrist. Make sure the wrist guard directly touches the skin. When the strap is secured to your wrist, attach the alligator clip on the other end of the wrist guard to a grounded surface. To insure against ground faults, wear a wrist guard that contains a 1MΩ (one megohm) resistor.
- Always store modules in their original packaging.
- Never operate the PowerHub 6000 with exposed slots or daughter card positions. Operating the unit without cover plates installed over unused openings voids the warranty.
- Never remove a module from its protective packaging or from a chassis until you, the chassis, and the work surface are properly grounded. If the work surface is metallic, you can ground it by attaching a wire from the surface to the electric ground in the building. If the work surface is not metallic, use a ground-conductive rubber mat as your work surface.

> **NOTE**: Low humidity levels can increase the danger of electrostatic discharge. Use extra caution if the PowerHub 6000 is in a low-humidity environment.

- After you, the chassis, and any loose components are grounded, touch the chassis or work surface containing the component, ***before*** touching the component itself. In this way, you neutralize any charge before it can damage the component. Note that you and the chassis or other surface ***must*** be grounded for this to be effective.
- Handle the modules only by their edges. Never directly touch components on the modules.

4.1.3 Connector Pins

Use care when handling modules that have connector pins. Some modules, such as the daughter cards that fit on the Packet Engine, contain many small pins that plug into headers in other modules. These pins can be easily bent if you drop or slide the module, or if you mis-align or use excessive force to seat the module's pins in a header.

Pin damage caused by mishandling is not covered by your warranty. If you do accidentally bend a pin, you sometimes can prevent the pin from becoming broken by carefully bending it back into position before attempting to seat the module.

4.1.4 Care of Fiber-Optic Systems and Cables

If you plan to install a 10Base-FB or a 10Base-FL NIM or EMA, or one of the optional FDDI daughter cards, you need to read this section; otherwise, skip this section.

In addition to the general precautions discussed in Section 4.1, fiber-optic systems require some additional precautions. Always use care when connecting fiber optic cables. Although they look like standard copper cables, they are delicate. Avoid repeated sharp bending of fiber optic cable since it can cause micro-cracking of the glass fiber. Be particularly careful of the open ends of the uncovered connectors. Make sure that the connector surfaces are not dragged along the floor or dropped onto hard or abrasive surfaces. We recommend that you keep the factory supplied dust covers on all unused fiber connectors and PowerHub optical components.

4.2 REQUIREMENTS

Before you begin installing your PowerHub 6000, make sure the area in which you plan to install it meets the following environmental and power requirements.

4.2.1 Environment

The PowerHub 6000 is designed to operate within ambient temperatures of 0° to 40° C (32° to 104° F) and at 10 to 90 percent relative humidity, non-condensing. The corresponding storage requirements are –20° to 55° C (–4° to 131° F) and 90 percent maximum relative humidity, non-condensing.

To ensure adequate air flow for cooling, the PowerHub 6000 requires at least 3" clearance on each side. Operating the hub without adequate clearance for cooling may void the warranty.

4.2.2 Power

Each PowerHub 6000 requires one power supply to run, but contains a bay for a second (redundant) power supply.

The power supply requires an input voltage of 85 - 132 VAC (or 180 - 264 VAC at 47 - 60 Hz). The power supply runs on a standard 110-volt AC or a 220-volt AC power source, and automatically detects the input voltage. You never need to set jumpers or switches when setting up for an alternative power source.

The power supply draws a maximum of 5A at 110 volts or 2.5A at 220 volts. Make sure the area in which you plan to install the hub provides an AC outlet meeting these specifications. The PowerHub 6000 is shipped with a three-wire power cord that matches the power receptacles used in the destination country.

4.2.2.1 Redundancy

The redundant power supply provides a backup if the primary power supply ever fails. If the PowerHub 6000 does not have a redundant power supply and the primary power supply fails, the hub shuts down, interrupting network service.

See Section 5.1.1 on page 73 for instructions on installing a power supply.

4.3 INSTALLATION

To install the PowerHub 6000, perform the following procedures:

- Install the chassis, either on a tabletop or in an equipment rack. (See Section 4.3.1.)
- Attach a “dumb” management terminal or modem (configured for 9600 baud) to the Packet Engine. (See Section 4.3.2 on page 55.)
- Power on the hub. (See Section 4.3.3 on page 59.)
- If needed, configure the software boot source (Flash Memory Module, TFTP server, or both; the default is Flash Memory Module). (See Section 4.3.4 on page 60.)
- Boot the software. (See Section 4.3.3 on page 59.)
- Run the diagnostic self-tests. (See Appendix A.)
- Attach your network segments to the hub. (See Section 4.3.6 on page 65.)
- Make a backup copy of the software. (See Section 4.3.6 on page 65.)

The following sections contain procedures for performing each of these tasks.

4.3.1 Installing the Chassis

This section describes how to install the PowerHub 6000 chassis.

The PowerHub 6000 chassis is 5-1/4” tall (5-1/2” when the rubber feet are installed), 17-1/2” deep, and 17” wide without its rack-mounting ears. You can install the PowerHub 6000 in any one of the following ways:

- On a tabletop.
- In a standard 19” equipment rack using one of the following:
  - Front-mount brackets, for closed-frame racks.
  - Center-mount brackets, for open-frame racks.

4.3.1.1 Tabletop

The PowerHub 6000 is compact enough to be mounted on a tabletop. To install the chassis on a tabletop, prepare enough space on the tabletop to allow at least 3” of open space around the sides and in front of the chassis, then place the chassis in the space you prepared.

4.3.1.2 Closed-Frame Rack

The PowerHub 6000 can be mounted in a standard 19” closed-frame rack using the small angle brackets and the screws supplied in the Brackets Package shipped with your hub.

For this type of installation, you need:

- Two closed-rack mounting brackets. These mounting brackets are smaller than the open-frame brackets and each have four screw holes for the PowerHub chassis and two screw holes for the rack.
- Four 12-24 1/2” pan-head screws, two for each bracket.
- Eight 6-32 3/8” flat-head screws.
- A #2 Phillips-head screwdriver.
- A flat-head screwdriver.

To install the chassis in a closed-frame rack:

(1) Carefully place the PowerHub chassis on its side, so that you can freely access the four rubber feet on the bottom.

(2) Use the #2 Phillips-head screwdriver to remove the rubber feet.

(3) Use the #2 Phillips-head screwdriver to remove the four 6-32 3/8” flat-head screws from one side of the front of the chassis.

(4) Align one of the closed-frame brackets over the screw holes, making sure the flange is behind the front of the chassis.

(5) Use the #2 Phillips-head screwdriver to insert the four 6-32 3/8” flat-head screws. Do not over-tighten the screws; they should be hand tight.

(6) Repeat steps 3–5 on the other side of the chassis for the other bracket.

(7) When both brackets are installed, carefully lift the chassis into the rack, align the screw guides over the holes in the rack, then use the flat-head screwdriver to insert the four 12-24 1/2” pan-head screws provided. The screws go into the mounting holes in the brackets.

4.3.1.3 Open-Frame Rack

The PowerHub 6000 can be center-mounted in a standard 19” open-frame rack using the large angle brackets supplied. For this type of installation, you need the following items:

- Two open-rack mounting brackets. These mounting brackets are larger than the closed-frame brackets and have twelve screw holes for the PowerHub chassis and four screw holes for the rack.
- Four 12-24 1/2” pan-head screws, two for each bracket.
- Eight 6-32 3/8” flat-head screws. (These are already in the chassis.)
- A #2 Phillips-head screwdriver.
- A flat-head screwdriver.

To install the chassis in an open-frame rack:

(1) Carefully place the PowerHub chassis on its side, so that you can freely access the four rubber feet on the bottom.

(2) Use the #2 Phillips-head screwdriver to remove the rubber feet.

(3) Use the #2 Phillips-head screwdriver to remove the two 6-32 3/8” flat-head screws from one side of the front of the chassis.

(4) Align one of the open-frame brackets over the screw holes, making sure the flange is behind the front of the chassis.

(5) Use the #2 Phillips-head screwdriver to insert the four 6-32 3/8” flat-head screws. Do not over-tighten the screws; they should be hand tight.

(6) Repeat steps 3–5 on the other side of the chassis for the other bracket.

(7) When both brackets are installed, carefully lift the chassis into the rack, align the screw guides in the bracket flanges with the screw holes in the rack, then use the flat-head screwdriver to insert the four 12-24 1/2” pan-head screws provided. The screws go into the mounting holes in the brackets.

4.3.2 Connecting the Management Terminal or Modem to the Hub

You attach your management terminal or modem to the PowerHub 6000 using an RS-232 cable. The PowerHub 6000 is shipped with an RS-232 cable assembly kit, which contains parts to assemble the cable. As shown in Figure 4–1, one end of the cable plugs into the PowerHub TTY1 port. The other end of the cable plugs into the DTE or DCE connector you assemble using the following procedure.

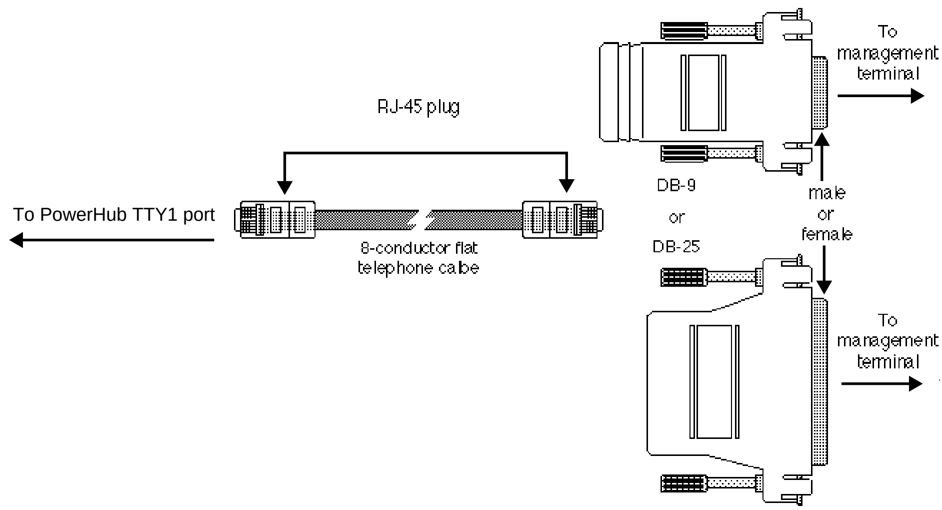


FIGURE 4–1 How to connect the PowerHub 6000 to your management terminal or modem.

> **NOTE**: The first time you log on to the hub, the modem or management terminal attached to TTY1 must be set to 9600 baud.

To connect a management terminal or modem to the PowerHub 6000:

(1) Determine the type of connector shell you need for the management terminal or modem side of the connection. For example, if you are using a terminal whose serial port is a female DB-9 (D-shaped, with nine pin holes), you need the male DB-9 connector shell.

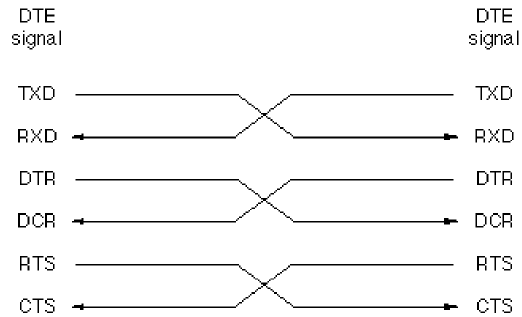
(2) Select the RJ-45 cable and the DB-9 or DB-25 connector shell that fits your terminal or modem serial port from the assembly kit.

(3) Using Table 4–2 and the appropriate figure from the following pages (Figure 4–2 through Figure 4–5) as guides, assemble the connector. For example, if you need a DB-9 connection to a PC DTE, and you want to incorporate the null-modem function into your cable, use the DB-9 and DCE columns of Table 4–2 and use Figure 4–4.

(4) Set the serial port on your modem or management terminal to 9600 baud.

(5) Plug the RJ-45 connector at one end of the supplied cable into the TTY1 port (see Figure 4–1).

(6) Plug the other RJ-45 connector on the RS-232 cable into the connector you have just assembled. Plug the DB-9 or DB-25 connector into the serial port on your management terminal or modem. If the management terminal or modem port is wired as DTE, use a null modem or some other means to swap the signal pairs in the RS-232 cable:



One way to do this is simply to wire the connector or management terminal as DCE.

TABLE 4–1 RS-232 pinout reference for the PowerHub TTY port.

TTY1 Signal	RJ-45 Pin No.	Direction
TXD	6	OUT
RXD	3	IN
GND	4	—
DTR	7	OUT
DCD	5	IN
RTS	8	OUT
CTS	1	IN

TABLE 4–2 RS-232 (TTY) pinout reference for terminal and modem.

RJ-45 Pin No.*	Terminal Signal (DTE)	DB-25 Pin No.	DB-9 Pin No.	Modem Signal (DCE)	DB-25 Pin No.	DB-9 Pin No.
6 (Yellow)	TXD	3	2	TXD	2	3
3 (Black)	RXD	2	3	RXD	3	2
4 (Red)	GND	7	5	GND	7	5
7 (Brown)	DTR	6	6	DTR	20	4
5 (Green)	DCD	20	4	DCD	8	1
8 (White)	RTS	5	8	RTS	4	7
1 (Blue)	CTS	4	7	CTS	5	8
2 (Orange)	n/a	n/a	n/a	DSR	6	6

* The colors refer to the wires inside the connector shells, not to the wires in the cable itself.

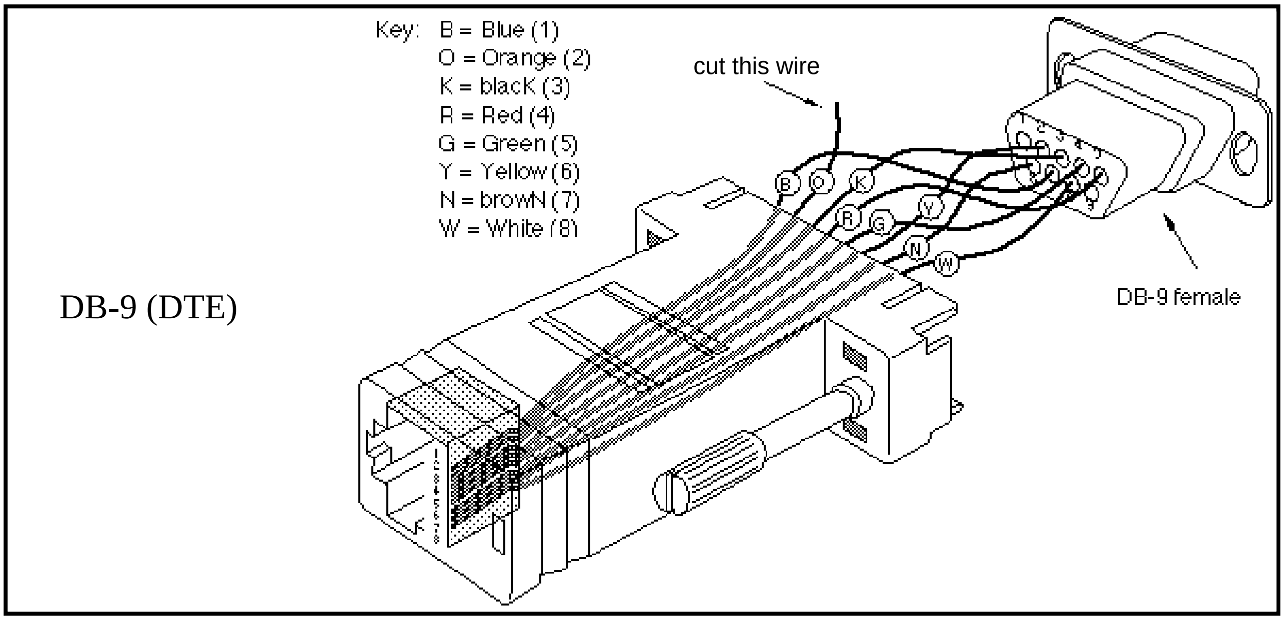


FIGURE 4–2 Pin assignments for DB-9 DTE (to terminal device) connector.

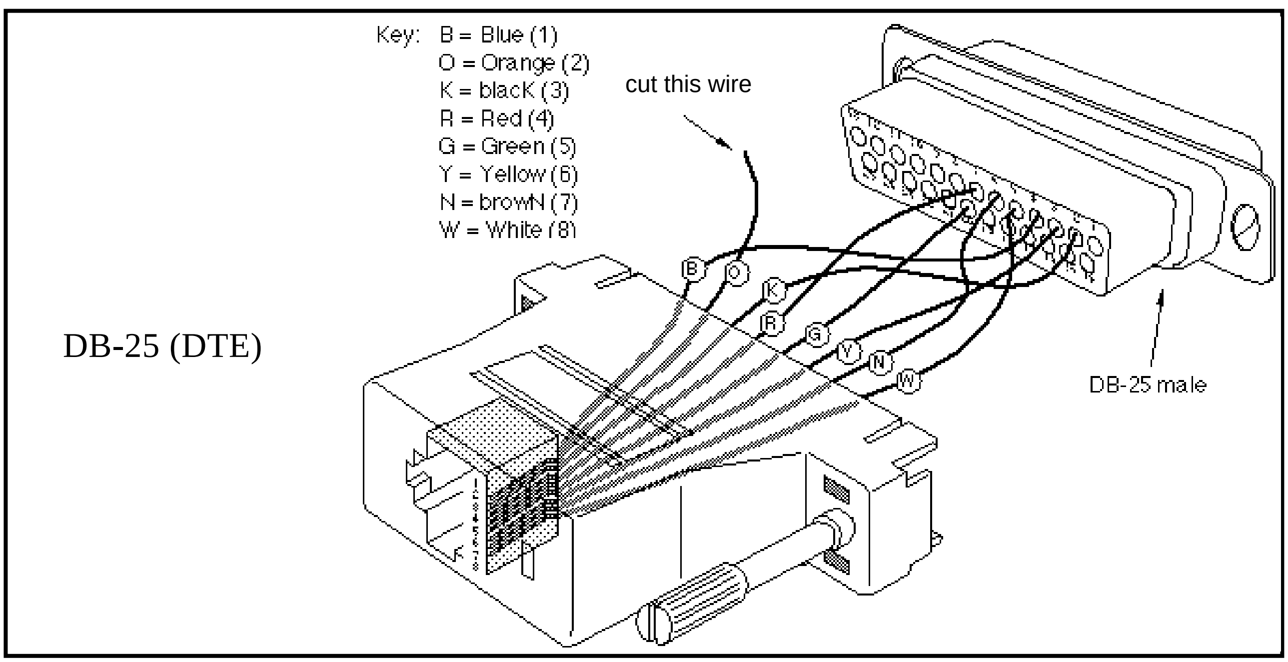


FIGURE 4–3 Pin assignments for DB-25 DTE (to terminal device) connector.

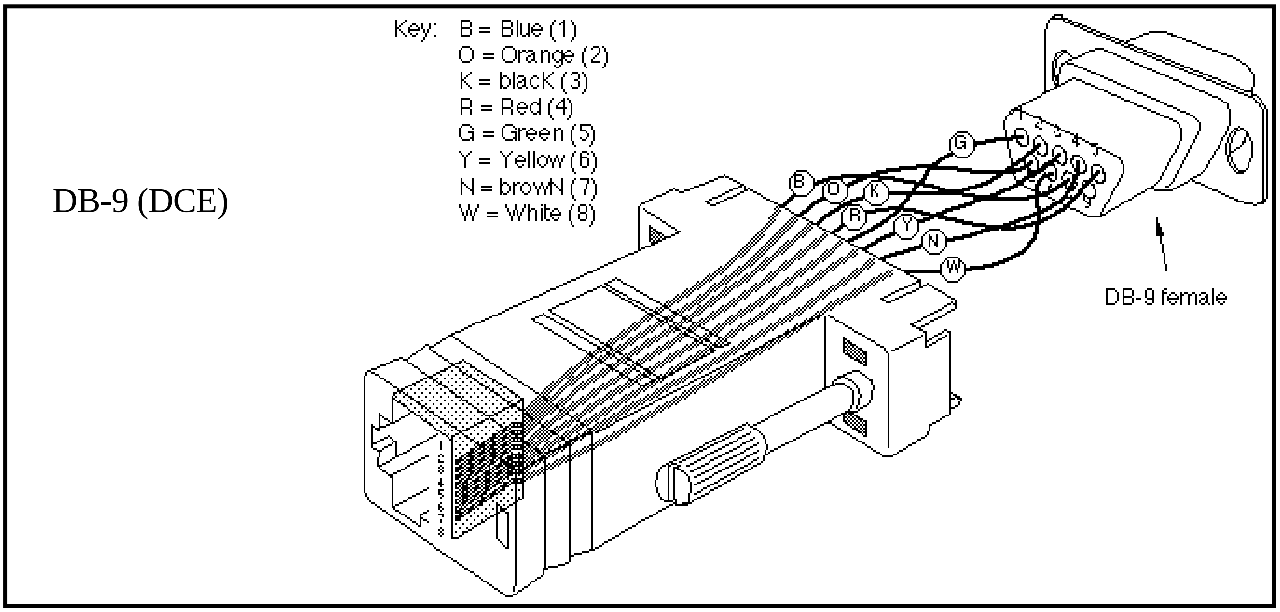


FIGURE 4–4 Pin assignments for DB-9 DCE (to modem device) connector.

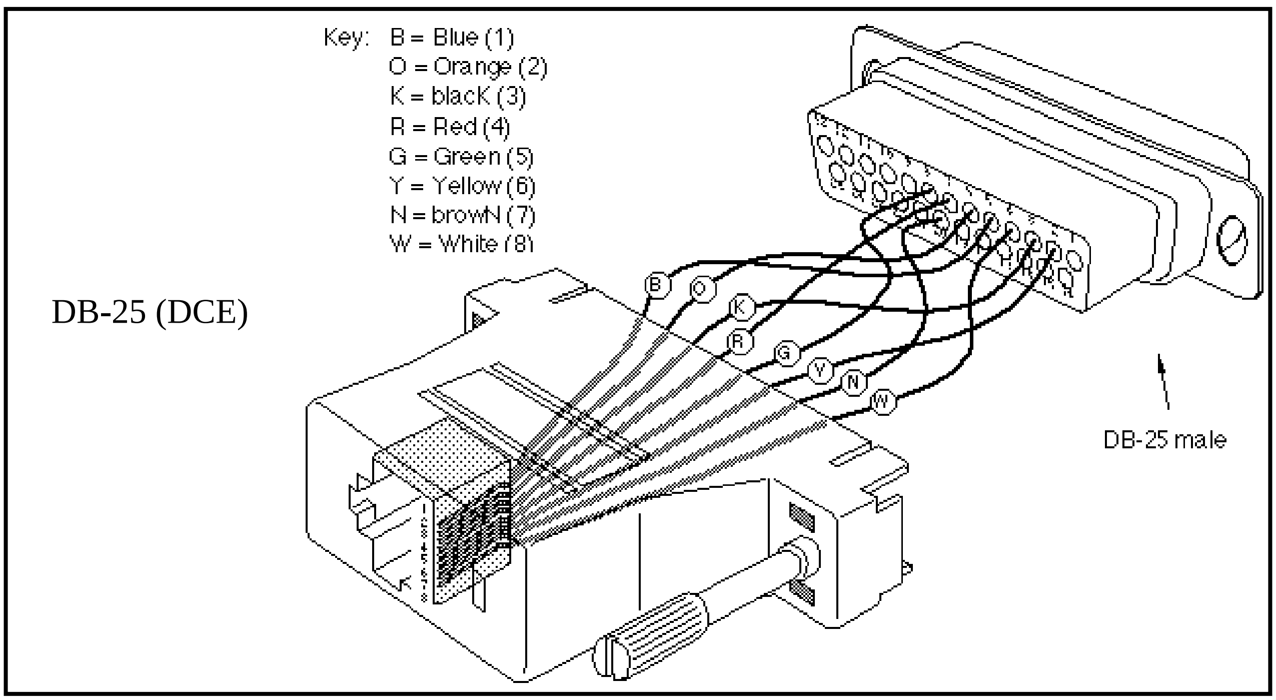


FIGURE 4–5 Pin assignments for DB-25 DCE (to modem device) connector.

4.3.3 Powering on the Hub

To power on the PowerHub 6000:

(1) Make sure the chassis contains at least the primary power supply. The primary power supply goes in the power supply bay labeled PS1.

(2) Make sure all unused slots and bays in the chassis are covered with their cover plates.

(3) Plug the power supply cable into the power supply, then into a grounded circuit capable of supplying the required amount of current.

(4) Switch on the power supply. If the chassis contains two power supplies, switch them both on at the same time. Note that the second power supply is upside down relative to the primary supply. Make sure you are setting the switch on both supplies to the on position.

> **NOTE**: If your chassis contains two power supplies, make sure you always turn both of them on at the same time. Moreover, if you need to power down the PowerHub 6000, make sure you switch both power supplies off at the same time.

As soon as you turn on the primary power supply, the hub performs a lamp test and a power-on self test. During the lamp test, all the LEDs are illuminated for about five seconds. If any of the LEDs fails to light during the lamp test, repeat the procedure above. If the LED again fails to light, contact FORE Systems TAC.

During the power-on self-test, the hub displays messages similar to the following.

```
FORE Systems PowerHub 6000 Packet Engine
Prom version: 2.5 (s1.61) 1995.04.19 09:35
I-cache 16K OK
entering cached code
I-cache execution OK
D-cache 4K OK
SRAM 128K OK
DRAM .............7680K OK
Shared Memory ...2048K OK
Entering Monitor
FlashInit: found 2MB Flash Memory Module
Board Type: 6PE ,  CpuType: MCPU,  Instance: 1
Ethernet address: 00-00-ef-01-e4-80

(normal start)

Hit any key now to abort boot    :      [5]:
```

When the hub completes the lamp test and power-on self test, the Boot (B) LED on the Packet Engine flashes to indicate that the hub is attempting to boot the software.

If the boot process is successful, a command prompt such as the following is displayed:

```
1:PowerHub:main#
```

If you abort the boot process or an error occurs during booting, the software does not boot. Instead, the following prompt is displayed:

```
<PROM-6pe>
```

To resume the boot process, issue the following command at the `<PROM-6pe>` prompt:

`b [net|fm]`

where:

`net` Boots from the network. (You must configure the hub for netbooting.)

`fm` Boots from the Flash Memory Module.

If you do not specify a boot source, the boot order configured in NVRAM is used. If you have not configured a boot order, the system attempts to boot first from the Flash Memory Module, then from the network. If your chassis does not contain the Flash Memory Module and you have not yet configured for network booting, see Section 4.3.4 on page 60.

4.3.4 Configuring the Boot Source

You can configure the PowerHub 6000 to boot the software from either of the following sources:

- Flash Memory Module.
- BOOTP/TFTP server (network booting or “netbooting”).

The hub is configured at the factory to first attempt to boot from the Flash Memory Module, then from the network. You can configure the hub to use one method exclusively, or to try one method first, then the other method.

> **NOTE**: The procedures in the following sections help you configure a single PowerHub 6000 for netbooting. If you want more detailed information about netbooting, or you want to configure multiple hubs for netbooting, read Appendix B before continuing with this section.

4.3.4.1 Specifying the Boot Source in NVRAM

You can configure the PowerHub 6000 to boot exclusively from the Flash Memory Module, exclusively from the network, or you can configure the hub to attempt both methods in the order you specify. When you configure the hub to try both methods, the hub always tries the primary method first, but if this method fails, it tries the secondary method. The hub is configured at the factory to attempt to boot from the Flash Memory Module first, then from the network.

To specify the boot source:

(1) If you have not already done so, power on the hub.

(2) Type the following command at the runtime command prompt (or the `<PROM-6pe>` prompt; the command can be issued at either prompt), then press Enter:

`nvram set bo` *`<value>`*

where:

`<value>` Specifies the boot order. You can specify one of the following:

- **`m`** Attempt to boot only from Flash Memory Module.
- **`n`** Attempt to boot only from the network.
- **`mn`** Attempt to boot from the Flash Memory Module first. If this attempt is unsuccessful, then boot from the network. (This is the default.)
- **`nm`** Attempt to boot from the network first. If this attempt is unsuccessful, then boot from the Flash Memory Module.

> **NOTE**: If you specify both boot sources, make sure the `cfg` (default configuration) file in the Flash Memory Module and its counterpart that you install on the TFTP server match. Otherwise, the hub's configuration might differ depending upon the boot source.

4.3.4.2 Configuring for Network Booting

The steps required to configure for netbooting the PowerHub 6000 differ depending upon whether the hub and the boot server are on the same subnet or on different subnets. The subnet can be a single segment or multiple segments connected by bridges.

- If the PowerHub 6000 and the boot server are on the same subnet, use the procedure in Section 4.3.4.3 on page 62 to implement point-to-point netbooting.
- If the PowerHub 6000 and the boot server are on different subnets, use both the procedure in Section 4.3.4.3 on page 62 and the procedure in Section 4.3.4.4 on page 64 to implement cross-gateway netbooting.

Unless you specified the boot source in the procedure in Section 4.3.4.1 on page 61 as **`m`** (Flash Memory Module only), you must perform one of the procedures in the following sections.

> **NOTE**: The following procedures recommend specific pathnames for storing the PowerHub 6000 system software, boot definition, and configuration files. Use pathnames that are legal on your TFTP server, but make sure the pathnames are used consistently. For example, you will need to copy the `bootdef` (boot definition) file onto the TFTP server and specify the pathnames in the `bootdef` file. The pathnames in the `bootdef` file must match those specified on the server. Alternatively, you can install all the files in the TFTP home directory.

4.3.4.3 Point-to-Point

To configure the PowerHub 6000 and the boot server for netbooting the hub, you must:

(1) Configure the TFTP server as follows:

- Copy the system software image file (`6pe`) and boot definition file (`bootdef`) from a software diskette onto the server. If you plan to install them in the TFTP home directory, skip the next bulleted item.
- If you want to set up separate TFTP subdirectories for the software image file, boot definition file, and configuration file, then create the following subdirectories:

  `fore/ph/images/6-2.6.3.0`

  `fore/ph/configs`

  Copy the image file (`6pe`) into the `6-2.6.3.0` subdirectory. Copy the boot definition file (`bootdef`) and the configuration file (when you create one) into the `configs` subdirectory. (Section 4.3.8.2 on page 68 describes how to create a configuration file and save it on the TFTP server.)
- Edit the boot definition file (`bootdef`) you copied onto the server to contain the pathname and file name of the configuration file and the pathname and file name of the `6pe` file. Here is an example.

```
%vstart 1
fore/ph/configs/0000EF014A00.cfg     c
fore/ph/images/6-2.6.3.0/6pe         m
%vend 1
```

In this example, the PowerHub 6000's MAC-layer hardware address is used for *`<file-name>`* in the configuration file name. Note that your hub's MAC-layer hardware address is unique (see the label on the front of the chassis). The procedures in Section 4.3.8 on page 67 describe how to save your PowerHub configuration into a configuration file. When you save the configuration to the TFTP server, make sure you save the file under the name you specify here in the boot definition file.

> **NOTE**: If your system contains the FDDI daughter card, make sure you edit the appropriate boot definition file: `bootdef.fdi`.

(2) Configure the BOOTP server or the Packet Engine's NVRAM to contain the following information. If you need to configure only one PowerHub system, you might want to configure all these values into NVRAM. By doing so, you avoid needing to configure the BOOTP server. However, if you plan to configure many PowerHub systems for netbooting, we recommend configuring these values on the BOOTP server instead.

- Client hub's IP address.
- Client hub's subnet mask.
- Gateway's IP address (if the client hub and server are on different subnets).
- TFTP server's IP address.
- Name of the boot definition file (often called `bootdef`) you plan to use to boot the client hub. You install this file on the TFTP server, but specify the name on the BOOTP server or in NVRAM. Note that the boot definition file is neither the image file (`6pe`) nor a configuration file (such as `cfg`).

To configure these values on your BOOTP server, see the documentation for your BOOTP server.

To configure these values in NVRAM, issue the following commands on the PowerHub 6000. You can issue these command from the `<PROM-6pe>` prompt or the runtime prompt:

nvram set myip *<value>*

where:

<value>	Is the PowerHub 6000's IP address. Specify the address in dotted decimal notation (ex: 147.128.16.1).

nvram set mysm *<value>*

where:

<value>	Is the PowerHub 6000's IP subnet mask. Specify the mask in dotted decimal notation (ex: 255.255.255.0).

nvram set gwip *<value>*

where:

<value>	Is the gateway router's IP address. Specify the address in dotted decimal notation (ex: 147.128.16.2).

nvram set fsip *<value>*

where:

<value>	Is the TFTP server's IP address. Specify the address in dotted decimal notation (ex: 147.128.16.3).

nvram set netbdfile *<value>*

where:

<value>	Is the name of the boot definition file on the TFTP server. Specify a name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `fore` and this directory is specified as the TFTP home directory, do not specify `fore` as part of the file name.

(3) Connect the PowerHub 6000 to the boot server (or intervening gateway) by attaching a network cable from the server to a segment on the PowerHub 6000.

(4) If the chassis contains BNC segments, make sure they are properly terminated. To terminate a BNC segment, set the switch on the segment to the terminated (T) position.

4.3.4.4 Cross-Gateway

Cross-gateway netbooting refers to a netbooting configuration in which the PowerHub 6000 (boot client) and the boot server are separated by a gateway router. The configuration steps for cross-gateway netbooting include the steps for point-to-point netbooting. In addition, configuring for cross-gateway netbooting requires one of the following steps:

- If the gateway has a boot helper service, configure the gateway to help BOOTP packets sent by the client PowerHub 6000 to reach the BOOTP server.[1] If the gateway is another PowerHub Intelligent Switching Hub, use the **ip add-helper** command on the gateway to configure a helper address.
- If the gateway does not contain a boot helper service, configure the following values in the client PowerHub 6000's NVRAM:
  - Client hub's IP address.
  - Client hub's subnet mask.
  - Gateway's IP address (if the client hub and server are on different subnets).
  - TFTP server's IP address.
  - Name of the boot definition file.

  The commands for configuring these values are described in Step 2 in the procedure in Section 4.3.4.3 on page 62.

  To configure for cross-gateway netbooting:

(1) If you have not already performed all of the steps in Section 4.3.4.4 on page 64, do so now.

(2) If the gateway that separates the PowerHub 6000 from the TFTP server is another PowerHub Intelligent Switching Hub, make sure the segment on the gateway hub that connects to the PowerHub 6000 is configured as an IP interface. If the gateway hub's segment does not have an IP interface, configure one using the following command on the gateway PowerHub system:

```
ip add-interface|ai <seg-list> <IP-addr>
    [<subnet-mask> [br0|br1]
    [[cost <cost>]]
```

The command arguments are described in Step 3 in Section 4.3.8.2 on page 68.

1. Boot helper services are sometimes called IP helper services or UDP helper services. The PowerHub software contains a feature called *IP Helper* that provides boot helper services.

(3) Issue the following command on the gateway PowerHub system (not on the PowerHub 6000 you are installing):

ip add-helper|ah *<seg-list>***|all** *<IP-addr>*

where:

*<seg-list>***|all**	Specifies the segment(s) that connect the gateway hub to the PowerHub 6000. You can specify multiple segments or **all** for all segments. If the segment list is set to **all**, then the IP address is assigned to all valid segments in the system.
<IP-addr>	Specifies the helper address. You must specify the IP address of the TFTP server (not the gateway hub or the PowerHub 6000) as the helper address.

4.3.5 Running Additional Self-Tests

The PowerHub 6000 is shipped with test scripts that let you verify the complete functional operation of the system at maximum performance. These self-tests are in addition to the lamp test and power-on self-test the hub performs each time you power it on. You do not need to run these additional self-tests each time you power on.

The additional self-tests should be run only when the hub is not carrying live network traffic. Also, one of these self-tests requires that you first remove all network segment cables and install loopback cables. Accordingly, you might want to run these self-tests now, before attaching your network cables to the hub.

See Appendix A for information on running the self-tests.

4.3.6 Attaching Network Segments

To attach your network segments to the PowerHub 6000, simply plug the segment cables into the appropriate segment connectors on the hub. Make sure to set the termination switch on all BNC segments (if the hub contains any) to `T` (terminated) or `U` (unterminated) as required by your network configuration.

4.3.7 Enabling Automatic Segment-State Detection

As described in Section 1.2.12 on page 15, the automatic segment-state detection feature monitors the state (up or down) of a segment attached to the hub. If the segment state changes, the feature automatically enables or disables bridging and routing on the segment and marks the change in table displays.

The default setting for the automatic segment-state detection feature differs depending upon the segment type. Table 4–3 lists the default setting for each segment type.

TABLE 4–3 Default setting for automatic segment-state detection.

Segment Type	Default
10Base-FB	Enabled
10Base-FL	Enabled
10Base-T (UTP)	Enabled
100Base-FX	Enabled
100Base-T4	Enabled
100Base-TX	Enabled
AUI	Disabled
BNC/BNCT	Disabled
FDDI	Enabled
MAU	Enabled

After you attach your network segments, you must enable the feature on all UTP segments before attempting to use the PowerHub 6000. See Section 9.5.4 on page 167 for a complete description of the automatic segment-state feature.

To enable automatic segment state detection, issue the following command:

autoportstate|aps [*<seg-list>***|all enl|dis [***<threshold>***]]**

where:

*<seg-list>***|all**	Specifies the segment(s) for which you want to enable or disable automatic segment-state detection.
enl|dis	Specifies whether you are enabling or disabling automatic segment-state detection on the specified segments.

NOTE: You must enable automatic segment-state detection on all 10Base-T segments. The Ethernet controllers refuse to transmit packets on any segment that does not have a “good” link status. As a result, buffers can become “stuck” on the output queue of 10Base-T segments that do not have a “good” link status. This can adversely affect the performance of the rest of the hub. By enabling automatic segment-state detection, you can prevent buffers from being enqueued on the segments, and allow any enqueued buffers to be recovered if the segments go down.

<threshold>	For AUI segments, specifies the “loss-of-carrier threshold”; that is, the number of times a loss of carrier must be detected in a 1-second period for the segment to be considered down and therefore be disabled by the software. For BNC segments, specifies the “idle period threshold”; that is, the number of seconds during which the segment must remain idle to be considered down and therefore be disabled by the software.

The default is **10** for AUI, or **5** for BNC. This argument does not apply to other media types, such as 10Base-T and 10Base-FL.

> **NOTE**: The threshold setting applies only to AUI and BNC segments.

4.3.8 Saving the PowerHub Configuration

When you use software commands to change the PowerHub 6000 configuration, the hub does not retain these changes after you power down or reboot the system. (However, changes made to the NVRAM are saved.) You can preserve configuration changes by saving them to a configuration file.

When you boot the PowerHub 6000, the software looks for a configuration file on the device you specified as the boot source. When the configuration file is read by the hub, the configuration changes you saved in the file are reinstated on the hub. The configuration file name is specified in the boot definition file:

- If the software is booted from the Flash Memory Module, the `bootdef` file on the Flash Memory Module identifies a file name on the module. By default, the `bootdef` file identifies the name `cfg`. (You can change the default configuration name in the boot definition file by editing the file.)
- If the software is booted from a BOOTP/TFTP server, the boot definition file identifies a file named *`<file-name>`*`.cfg`, where *`<file-name>`* identifies the PowerHub system. Depending on how you configure for netbooting, you can specify a descriptive string or the hub's MAC-layer hardware address. See Section B.4 on page 238 for details.

> **NOTES**: If you plan to use both the Flash Memory Module and a boot server as boot sources, we recommend that you issue the commands described in both sections below. You can avoid potential problems by always ensuring that the configuration files on both boot sources match.

4.3.8.1 Flash Memory Module

To save the current configuration onto the Flash Memory Module:

(1) Boot the software, if you have not already done so. Following the boot messages, a command prompt similar to the following is displayed:

```
1:PowerHub:
```

> **NOTE**: The software must be successfully booted before you can save the configuration. You cannot save the configuration from the `<PROM-6pe>` prompt.

(2) Type the following command at the prompt, then press Enter:

`mgmt savecfg cfg`

This command saves the configuration into a file named `cfg`. You can specify any DOS-like file name. (See Section 9.6.1 on page 176 for information on file naming conventions.) If you specify a name that does not match the name in the `bootdef` file (the default is `cfg`), you must load the configuration manually each time you boot the software by using the **`mgmt readcfg`** command. (See Section 9.8.3 on page 190.)

4.3.8.2 TFTP Server

To save the PowerHub configuration file onto a TFTP server, you can do either of the following:

- Save the configuration file to the Flash Memory Module using the procedure in Section 4.3.8.1 on page 67. Use the ZMODEM commands in the boot PROM to transfer the files to a PC or Macintosh. (See Section 10.7.1 on page 207 for a description of the ZMODEM commands.) Copy the configuration file from the PC or Macintosh onto a diskette, then copy the file from the diskette to the server. This method requires more time than using TFTP, but does not require that you define an IP interface.
- Define an IP interface on the PowerHub segment attached to the TFTP server, then use the PowerHub TFTP software to save the configuration into a file on the server. This following procedure describes how to do this.

(1) Boot the software, if you have not already done so. Following the boot messages, a command prompt similar to the following should be displayed:

```
1:PowerHub:
```

> **NOTE**: The software must be successfully booted before you can save the configuration. You cannot save the configuration from the `<PROM-6pe>` prompt.

(2) Attach a segment cable from the PowerHub system to the TFTP server.

(3) Define an IP interface on the segment that connects the hub to the server by issuing the following command.

ip add-interface|ai *<seg-list> <IP-addr>*
[*<subnet-mask>* **[br0|br1]**
[[cost *<cost>***]]**

where:

<seg-list>	Specifies the segment(s) to which you are assigning the IP address. You can specify a single segment, a comma-separated list of segments, a hyphen-separated range of segments, or **all** for all segments. If you specify multiple segments, you create a VLAN. (See Appendix D in the *PowerHub Software Manual, V 2.6* (Rev C).)
<IP-addr>	Specifies the IP address you want to assign to the specified segment(s). The IP address must be in dotted-decimal notation (four decimal numbers in the range **0**–**255** separated by dots).
<subnet-mask>	Specifies the subnet mask. If a particular network uses IP subnet addressing, then the subnet mask should be specified here using dotted-decimal notation. Otherwise, the system uses a default subnet mask equal to the “natural” subnet mask for the particular class of address.
br0|br1	Specifies the style of broadcast address on a segment-by-segment basis: • When you specify **br0**, the hub sends an “all-0s” broadcast. This means all bits in the host segment of the address are 0s. • When you specify **br1**, the hub sends a standard “all-1s” broadcast. This means all bits in the host segment of the address are 1s. The default is **br1**.
cost *<cost>*	Specifies an additional cost of using the subnet interface. This is the number of extra hops to the destination. The range is **1** through **14**. (The router decrements an IP packet’s time-to-live field at each hop.) The default is zero. When the hub reports this subnet using RIP, it adds the additional cost to the reported metric.

(4) Enable IP forwarding by issuing the following command: **ip set ifw**

(5) Type the following command at the prompt, then press Enter:

tftp svcfg [-h *<host>***]** *<remfile>*

where:

-h *<host>*	Specifies the IP address of the TFTP server. Unless you have already specified a default TFTP server using the **tftp set** command, you need to include this argument. For information on the **tftp set** command, see the *PowerHub Software Manual, V 2.6* (Rev C).

`<remfile>`	Specifies the configuration file name. Specify a name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `fore` and this directory is specified as the TFTP home directory, do not specify `fore` as part of the file name.

NOTES: This command saves the IP interface you defined in Step 3.

Some TFTP servers require that the remote file name exist on the server before you can write to that file name. If your server requires that the file name already exist, create a short file (named the same as your configuration file) on the server, then specify that file name for *<remfile>*.

Also, on some TFTP servers, including servers running Sun/OS 4.x, files that you overwrite on the server are not properly truncated. When you overwrite an existing file on the TFTP server, if the older version of the file is longer than the new file, the older version is not truncated properly by the server. As a result, the new version of the file contains part of the older version of the file.

If the configuration file name you specified in the boot definition file on the server is longer than eight characters, you can copy the file to the server using a DOS-like name, then rename the file on the server to match the file name you specified in the boot definition file.

5 Changing the Hardware

This chapter contains procedures for installing or removing the following PowerHub 6000 chassis components:

- Power supply. (See Section 5.1 on page 72.)
- Packet Engine. (See Section 5.2 on page 74.)
- Daughter card. (See Section 5.3 on page 77.)
- Flash Memory Module. (See Section 5.5 on page 81.)
- Packet Accelerator. (See Section 5.6 on page 85.)
- DRAM upgrade. (See Section 5.7 on page 88.)
- UMM (Universal Media Module). (See Section 5.8 on page 90.)
- EMA (Ethernet Media Adapter, a daughter card on the UMM). (See Section 5.9 on page 93.)
- Packet Channel backplane. (See Section 5.10 on page 99.)
- NIM (Network Interface Module). (See Section 5.11 on page 102.)

In addition, this section describes how to change the setting of the Lock Switch jumper. (See Section 5.12 on page 103.)

None of the procedures in this chapter are required for basic installation of the PowerHub 6000. For instructions on installing the PowerHub 6000, see Chapter 4.

All of these procedures can be performed while the chassis is in its equipment rack. However, if you would prefer to perform a particular procedure with the cover removed from the chassis, use the procedure in Section 5.13 on page 106 for removing and re-installing the chassis cover.

> **CAUTIONS**: Electrostatic discharges (shocks) can permanently damage the PowerHub 6000's electronic components. Carefully read all of Section 4.1 on page 49, "Safety Precautions," before you perform any of the procedures in this chapter.
>
> To prevent personal injury or damage to the PowerHub components, always turn off the power supplies and disconnect the power cables before performing these procedures. After turning off the power to the system, wait around thirty seconds for the cooling fans to stop before opening the chassis. The fans are located on the right side of the chassis.
>
> Some procedures require the use of a #2 Phillips-head screwdriver. Do not use a #1 Phillips-head screwdriver for these procedures; it can strip the screw heads. A #2 screwdriver is larger than a #1 screwdriver and can be identified by its tip, which is flat rather than pointed.

5.1 POWER SUPPLY

The following procedures describe how to install and remove a power supply. For reference, Figure 5–1 shows how a power supply is installed in the PowerHub 6000 chassis.

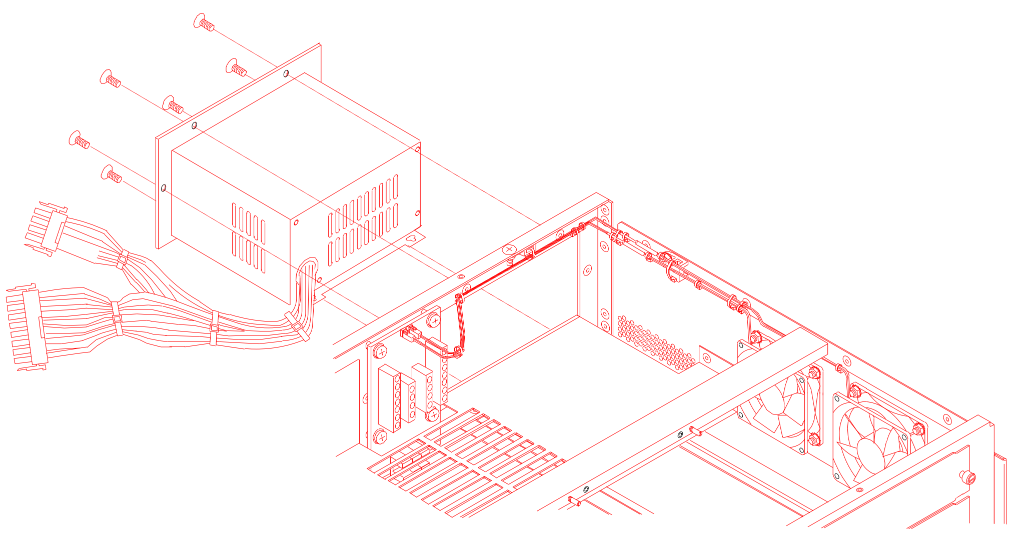


FIGURE 5–1 How a power supply fits into the PowerHub 6000 chassis.

5.1.1 Installing a Power Supply

For this procedure, you need a #2 Phillips-head screwdriver.

To install a power supply:

(1) If a coverplate is installed over the power supply bay, use the #2 Phillips-head screwdriver to remove the coverplate and store it in a safe place.

(2) Plug the two connectors into the matching sockets on the power backplane, as shown in Figure 5–2. The connectors are keyed and fit only one way onto the sockets. Make sure you plug the connectors into the sockets on the same side as the power supply.

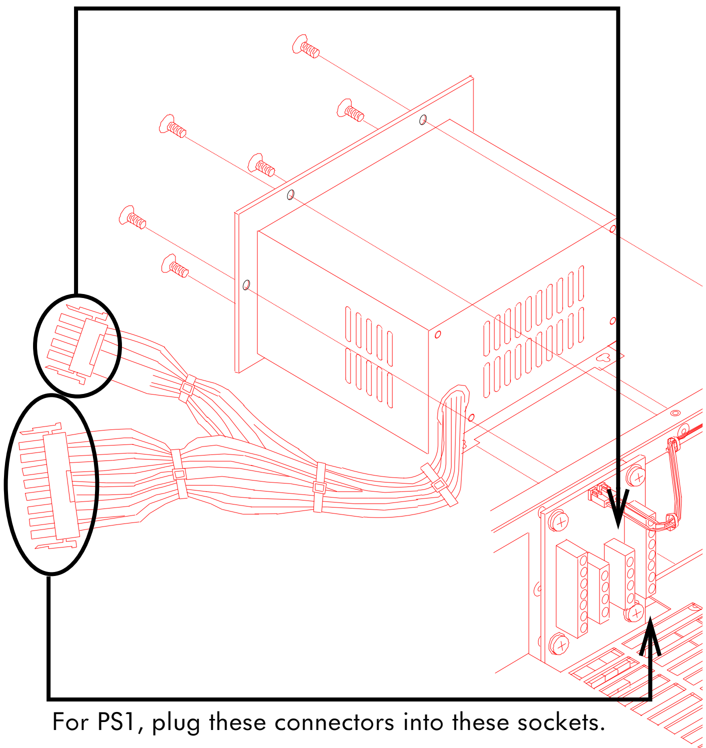


FIGURE 5–2 Where the power supply cables plug into the power backplane.

(3) After you plug the connectors into the appropriate sockets on the power backplane, insert the power supply all the way into the bay. Make sure the power supply cables are not accidentally pinched by the power supply. Note that the secondary power supply is installed upside down relative to the primary supply. Make sure you install the supply according to the orientation shown in Figure 5–3 on page 74.

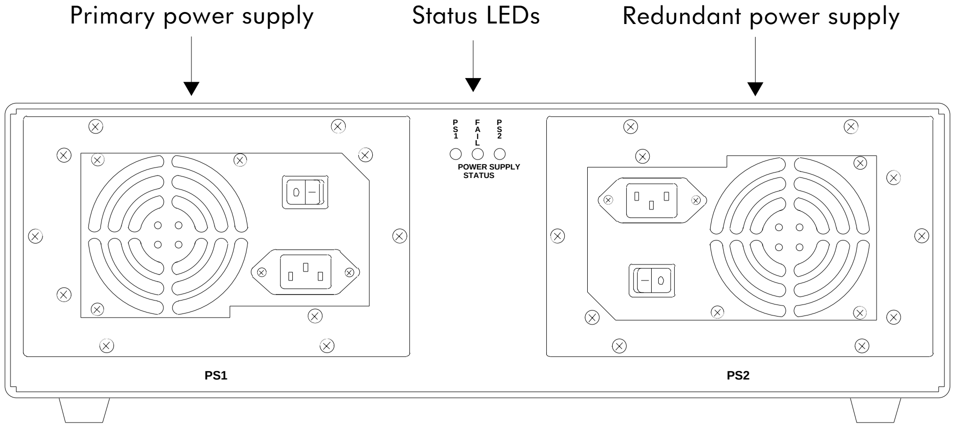


FIGURE 5–3 Rear view of PowerHub 6000 chassis, showing power supplies.

(4) Use the #2 Phillips-head screwdriver to insert the six screws that secure the power supply to the chassis, as shown in Figure 5–1.

5.1.2 Removing a Power Supply

For this procedure, you need a #2 Phillips-head screwdriver.

To remove a power supply:

(1) Remove the screws that secure the power supply to the chassis, as shown in Figure 5–1.

(2) Unplug the two connectors from the power backplane.

(3) Use the procedure in Section 5.1.1 on page 73 to install a new power supply. If you do not plan to install a new power supply, use the screws you removed to install a cover plate (part # 171-1213-0001) over the empty power supply bay.

5.2 PACKET ENGINE

The following procedures describe how to install and remove the Packet Engine.

5.2.1 Installing the Packet Engine

This procedure assumes that no UMM or NIMs are currently in the chassis. If the chassis contains the UMM or a NIM, carefully remove these modules using the appropriate procedures in this chapter:

- To remove a NIM, use the procedure in Section 5.11.1.
- To remove the UMM, use the procedure in Section 5.8.1.

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

To install the Packet Engine:

(1) If the Packet Engine contains the Packet Channel backplane, use the flat-head screwdriver to remove the coverplates from the middle and top slots in the chassis, if these coverplates are installed.

(2) Align the rear of the Packet Engine in the bottom of the chassis, then gently slide the Packet Engine into the slot.

(3) If the Packet Engine contains the Packet Channel backplane, use the #2 Phillips-head screwdriver to tighten the two thumbscrews that secure the backplane to the chassis.

(4) Use the flat-head screwdriver to tighten the screws on the front panel of the Packet Engine.

(5) If you plan to install the UMM or a NIM in the chassis, use the appropriate procedure in this chapter:

   - To install a NIM, use the procedure in Section 5.11.1.
   - To install the UMM, use the procedure in Section 5.8.1.

   Otherwise, use the flat-head screwdriver to replace any coverplates you removed. For an UMM or NIM, use coverplate part # 171-1207-0001.

NOTE: Make sure you install the appropriate cover plates over all unused slots, EMA positions, and power supply bays.

5.2.2 Removing the Packet Engine

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

To remove the Packet Engine:

(1) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Remove the NIM or UMM from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) If the Packet Engine contains the optional Packet Channel backplane, use the flat-head screwdriver to remove the coverplates, if present, from the middle and top slots in the chassis.

(4) If the Packet Engine contains the optional Packet Channel backplane, use the #2 Phillips-head screwdriver to loosen the two thumbscrews that secure the backplane to the chassis.

(5) Use the flat-head screwdriver to loosen the screws on the front of the Packet Engine.

(6) Pull the Packet Engine out of the chassis. You might need to lift the front of the Packet Engine slightly to avoid catching the bottom on the inside of the chassis. If the chassis contains the Packet Channel backplane, it should remain attached to the Packet Engine.

(7) If you plan to add or remove components on the Packet Engine, or change the setting of a jumper, use the appropriate procedures in this chapter to do so.

> **NOTE**: Make sure you install the appropriate cover plates over all unused slots, EMA positions, and power supply bays.

5.3 DAUGHTER CARD

The following procedures describe how to install and remove a daughter card on the Packet Engine.

5.3.1 Installing a Daughter Card

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #1 Phillips-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- A grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–4 shows where a daughter card is installed on the Packet Engine. Refer to this figure as you perform the following procedure.

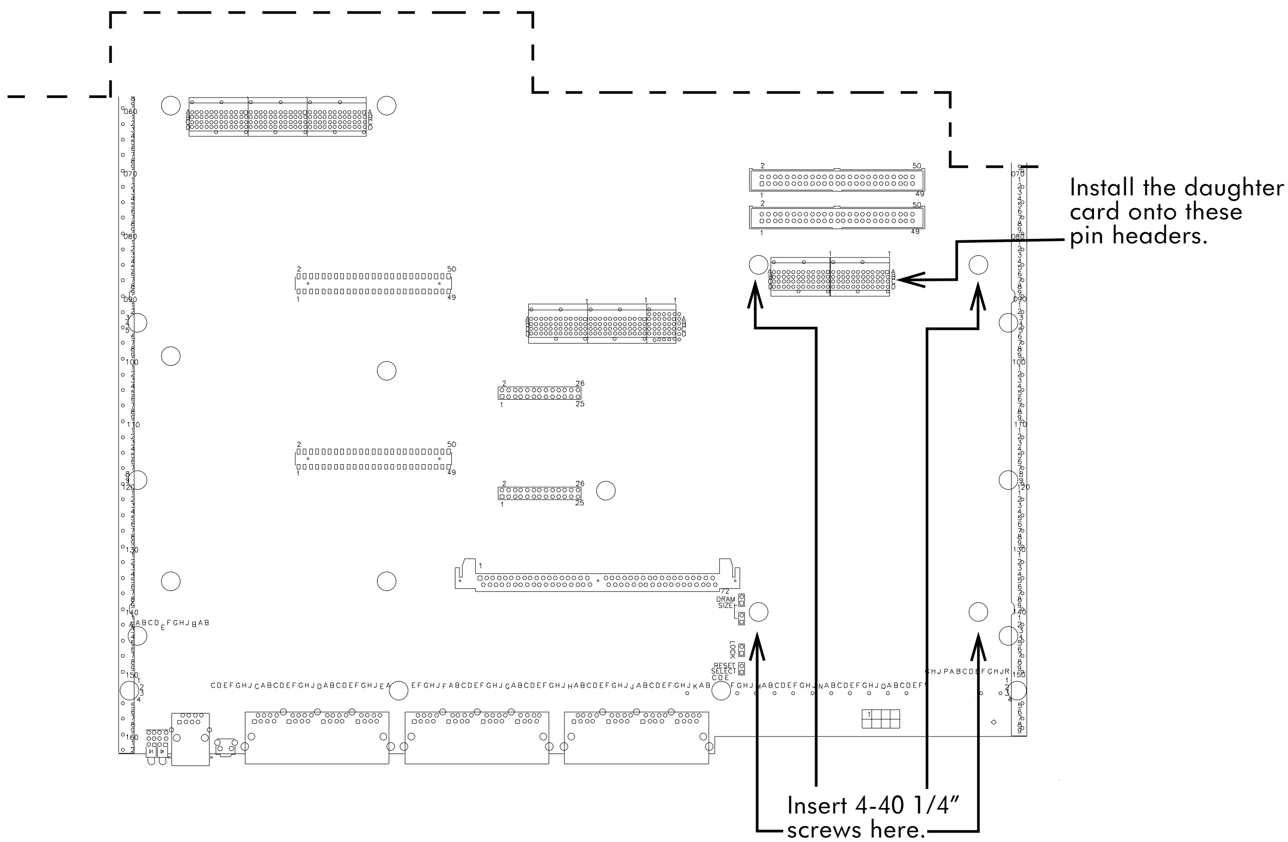


FIGURE 5–4 Where the daughter card is installed.

To install a daughter card:

(1) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Remove the NIM or UMM from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) Remove the Packet Engine. (See Section 5.2.2.)

(4) Holding the daughter card by its edges, turn the card so that it is component-side down, with the segment connector(s) facing the front of the chassis.

(5) Tilt the segment connector(s) down slightly, then insert the connector(s) into the corresponding opening in the Packet Engine front panel.

(6) Carefully align the pin header on the daughter card over the corresponding pin socket on the Packet Engine, as shown in Figure 5–4. The header is properly aligned when the header edges are flush with the pin socket.

(7) Gently press down on the daughter card where the header is aligned over the pins on the Packet Engine, until the header is completely in the pin socket.

(8) Use the #1 Phillips-head screwdriver to insert the four 4-40 1/4” pan-head screws into the holes that align over the corresponding standoffs on the Packet Engine. The screw holes are on the top of the daughter card and screw into the standoffs on the Packet Engine.

(9) Use the #1 Phillips-head screwdriver to insert the two 4-40 1/4” pan-head screws on both sides of the daughter card’s segment connector(s), on the Packet Engine’s front panel.

(10) Reinstall the Packet Engine. (See Section 5.2.1.)

(11) Reinstall the coverplates, UMM, or NIMs you removed from the middle and top slots. (See Section 5.11.1 for instructions on installing a NIM or Section 5.8.1 on installing the UMM.)

> **NOTE**: Make sure you install the appropriate cover plates over all unused slots, EMA positions, and power supply bays.

If you installed a PowerHub 6000 FDDI daughter card that has a MIC, you can use the procedure in Section 5.4 on page 80 to ensure that the MIC is used only for the connection type you want.

5.3.2 Removing a Daughter Card

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #1 Phillips-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–4 on page 77 shows where the daughter card is installed on the Packet Engine. Refer to this figure as you perform the following procedure.

To remove a daughter card:

(1) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Remove the NIM or UMM from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) Remove the Packet Engine. (See Section 5.2.2.)

(4) Use the #1 Phillips-head screwdriver to remove the four 4-40 1/4” pan-head screws that secure the daughter card to the Packet Engine. These screws are on the top of the daughter card and Packet Engine and screw into standoffs on the Packet Engine.

(5) Use the #1 Phillips-head screwdriver to remove the two 4-40 1/4” pan-head screws that secure the daughter card to the Packet Engine front panel.

(6) Gently pull up on the end of the daughter card closest to the rear of the chassis to free the pin header from the socket on the Packet Engine. If the daughter card does not lift freely, gently rock it from side to side to loosen it from the pins.

(7) Holding the rear of the daughter card slightly higher than the front, gently pull the daughter card straight back to remove the segment connector(s) from the Packet Engine faceplate.

(8) Place the daughter card in its protective packaging.

(9) If you do not plan to install another daughter card, reinstall the Packet Engine. (See Section 5.2.1.)

(10) Reinstall the coverplates, UMM, or NIMs you removed from the middle and top slots. (See Section 5.11.1 for instructions on installing a NIM or Section 5.8.1 on installing the UMM.)

(11) If you do not plan to immediately install another daughter card, install a coverplate (part# 171-1326-0001) over the empty daughter card position.

> **NOTE**: Make sure you install the appropriate cover plates over all unused slots, EMA positions, and power supply bays.

5.4 CONFIGURING A MIC FOR A SPECIFIC CONNECTION

The FDDI MICs are versatile enough to accommodate any of the four types of FDDI connection. This makes attaching the FDDI daughter card to an FDDI ring, concentrator, or other device simple. You do not need to make any modifications to a MIC connector to use it for any of the four FDDI connection types.

However, depending upon your network configuration, you might choose to use a MIC for only a certain type of connection. In this case, you can use the small plastic inserts provided with your FDDI daughter card to prevent unwary users from accidentally using the connector for connection types other than the one you want.

> **NOTE**: The ST connectors used on single-mode FMAs also support any of the four types of FDDI connections. However, these connectors do not use plastic inserts.

To configure a MIC for a specific type of FDDI connection:

(1) Open the small plastic bag containing the four plastic inserts.

(2) Select the insert corresponding to the FDDI connection type you need. Each insert is labeled with A, B, M, or S.

(3) Slide the insert into the slot on the MIC connector until it snaps into place. This slot is located on the bottom of the MIC. When the insert is properly placed, the insert label is facing out and the front of the insert is flush with the front of the MIC.

(4) Repeat this procedure for each MIC you want to configure.

To remove an insert:

(1) Use a narrow object to press down the small tab that holds the insert in place.

(2) Pull the insert out of the connector.

(3) Repeat this procedure for each MIC.

5.5 FLASH MEMORY MODULE

The following procedures describe how to install and remove the Flash Memory Module on the Packet Engine.

5.5.1 Installing the Flash Memory Module

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #1 Phillips-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–5 shows where the Flash Memory Module is installed on the Packet Engine. Refer to this figure as you perform the following procedure.

> **WARNING**: Make sure you correctly position the Flash Memory Module onto the Packet Engine pins. The Flash Memory Module is correctly positioned when the screw hole in the module is aligned over the corresponding standoff on the Packet Engine, and the supplied 4-40 1/4” screw is inserted through the screw hole into the standoff.
>
> If the Flash Memory Module is not correctly positioned on the Packet Engine pins, the Packet Engine can become damaged when you apply power.

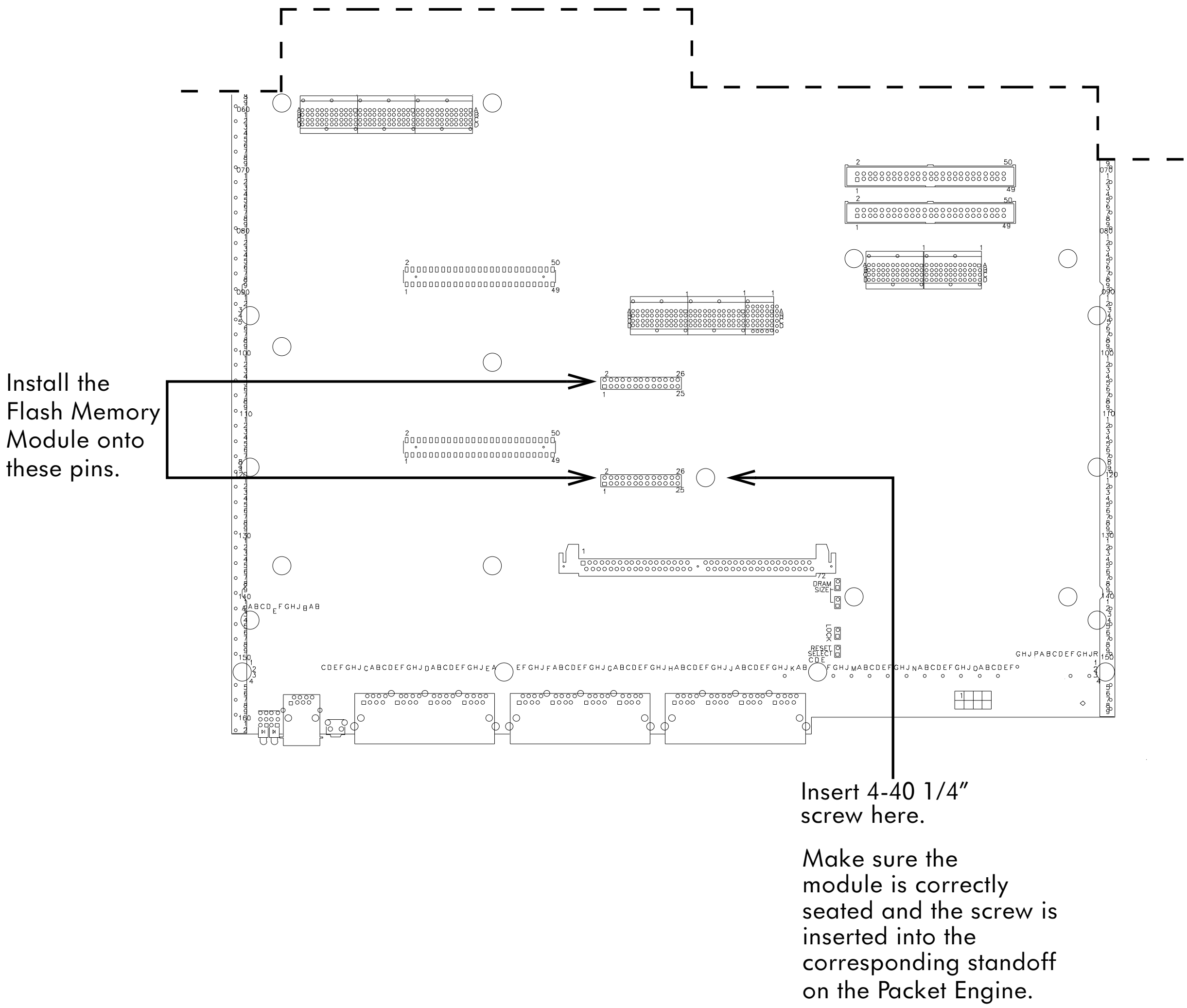


FIGURE 5–5 Where the Flash Memory Module is installed on the Packet Engine.

To install a Flash Memory Module.

(1) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Remove the NIM or UMM from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) If you did not remove the chassis cover, then remove the Packet Engine. (See Section 5.2.2.)

(4) Remove the Flash Memory Module from its protective packing.

(5) Holding the Flash Memory Module by its edges, align the two pin headers over the corresponding pins on the Packet Engine. Make sure the Flash Memory Module is oriented correctly over the pins. It is oriented correctly when the mounting hole in the lower right corner aligns with the standoff on the Packet Engine.

(6) Press down on the Flash Memory Module to seat the headers onto the pins.

(7) When the Flash Memory Module is completely seated, use the #1 Phillips-head screwdriver to insert the 4-40 1/4” screw into the corresponding standoff on the Packet Engine.

(8) If you removed the Packet Engine, reinstall it. (See Section 5.2.1.)

(9) Reinstall the NIM(s) or UMM you removed from the middle and top slots. (See Section 5.6.1 for instructions on installing a NIM or Section 5.8.1 on installing the UMM.)

(10) After you have completely finished the hardware changes, power on the unit and watch the boot messages for the line shown in bold type in the following example.

```
FORE Systems PowerHub 6000 Packet Engine
Prom version: 2.5 (s1.61) 1995.04.19 09:35
I-cache 16K OK
entering cached code
I-cache execution OK
D-cache 4K OK
SRAM 128K OK
DRAM ..............7680K OK
Shared Memory ...2048K OK
FlashInit: found 2MB Flash Memory Module
Board Type: 6PE ,  CpuType: MCPU,  Instance: 1
...remaining messages omitted for brevity.
```

If this line is present and shows the correct amount for the MB (2 or 4), the installation is a success. To verify the operation of the module, try using some of the file management commands described in Section 9.6 on page 176.

NOTE: If the file commands consistently return an error message such as <*command-name*>: `-65`, where *<command-name>* is the command you try to issue, you need to format the Flash Memory Module. To format the module, issue the following command: **`mgmt format fm`**

See Section 9.6.8 on page 181 for information on this command.

5.5.2 Removing the Flash Memory Module

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #1 Phillips-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

To remove a Flash Memory Module:

(1) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Remove the NIM or UMM from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) Remove the Packet Engine. (See Section 5.2.2.)

(4) Use the #1 Phillips-head screwdriver to remove the 4-40 1/4” screw that secures the Flash Memory Module to the Packet Engine.

(5) Gently pull up on the Flash Memory Module to free the pin headers from the pins on the Packet Engine. If the Flash Memory Module does not lift freely, gently rock it from side to side to loosen it from the pins.

(6) Store the Flash Memory Module in its protective packaging.

(7) If you removed the Packet Engine, reinstall it. (See Section 5.2.1.)

(8) Reinstall the NIM(s) or UMM you removed from the middle and top slots. (See Section 5.11.1 for instructions on installing a NIM or Section 5.8.1 on installing the UMM.)

5.6 PACKET ACCELERATOR

The following procedures describe how to install and remove the Packet Accelerator.

5.6.1 Installing the Packet Accelerator

For this procedure, you need the following:

- Grounded work surface.
- Packet Accelerator Upgrade Kit.
- A regular flat-head screwdriver.
- A #1 Phillips-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

To install a Packet Accelerator:

(1) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Remove the NIM or UMM from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) Remove the Packet Engine. (See Section 5.2.2.)

(4) Remove the Packet Accelerator from its protective packaging.

(5) Holding the Packet Accelerator by its edges, align the pin header over the corresponding pins on the Packet Engine (see Figure 5–6). Make sure the Packet Accelerator is oriented correctly. It is oriented correctly when the part number and version information are right-side up with respect to the front of the chassis. Figure 5–6 shows where the Packet Accelerator is installed on the Packet Engine.

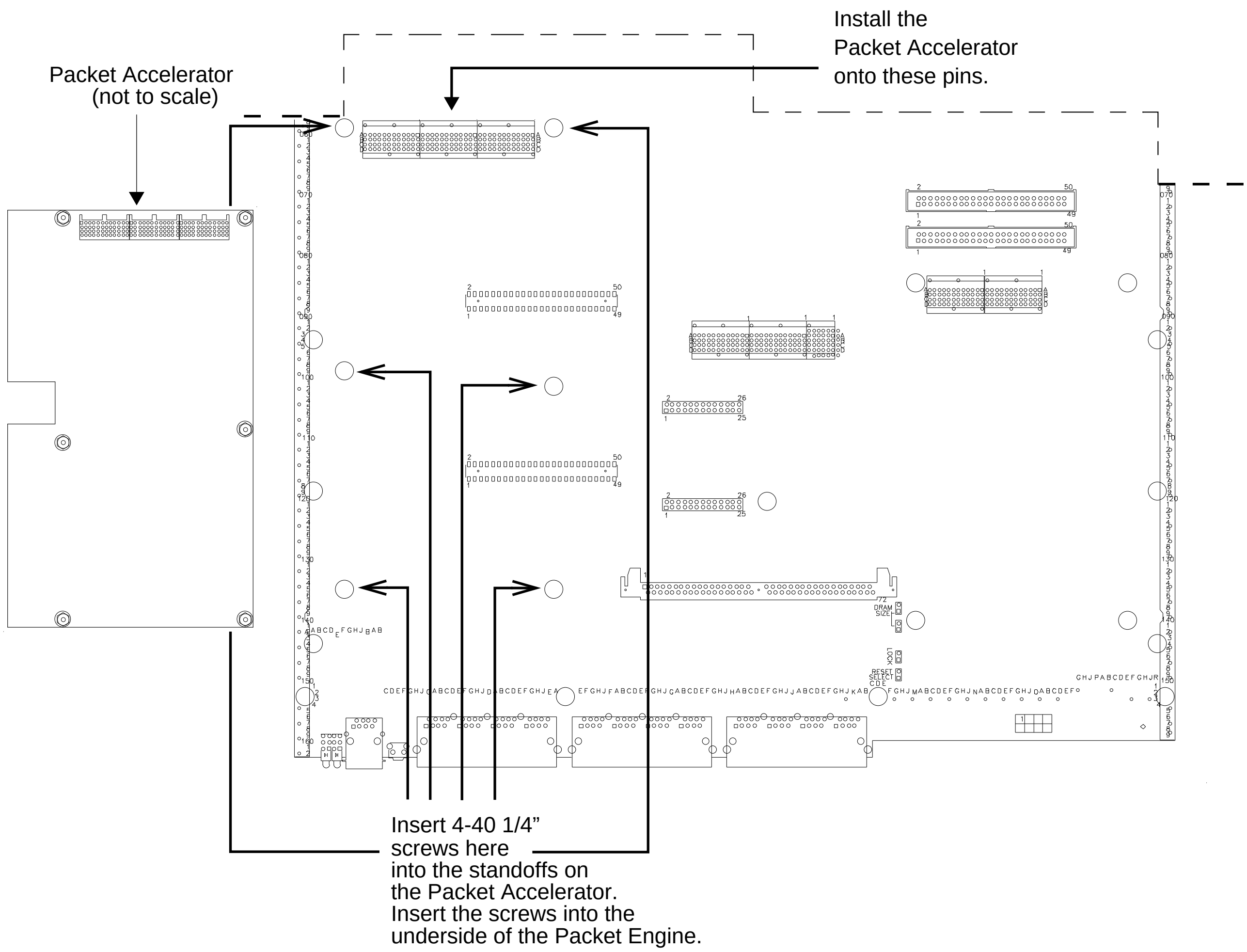


FIGURE 5–6 Where the Packet Accelerator is installed on the Packet Engine.

(6) Seat the Packet Accelerator onto the Packet Engine pins by pressing down on the rear of the Packet Accelerator, where the header is aligned over the pins. Be careful not to bend any pins.

(7) When the Packet Accelerator is completely seated on the pins, use the #1 Phillips-head screwdriver to insert the six 4-40 1/4” pan-head screws through the Packet Accelerator screw holes in the underside of the Packet Engine. These screw holes should line up over the standoffs on the Packet Accelerator.

(8) Reinstall the Packet Engine. (See Section 5.2.1.)

(9) Reinstall the NIM(s) or UMM you removed from the middle and top slots. (See Section 5.11.1 for instructions on installing a NIM or Section 5.8.1 on installing the UMM.)

5.6.2 Removing the Packet Accelerator

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #1 Phillips-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–6 on page 86 shows where the Packet Accelerator is installed on the Packet Engine. Refer to this figure as you perform the following procedure.

To remove a Packet Accelerator:

(1) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Remove the NIM or UMM from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) Remove the Packet Engine. (See Section 5.2.2.)

(4) Use the #1 Phillips-head screwdriver to remove the six 4-40 1/4” pan-head screws that secure the Packet Accelerator to the Packet Engine. These screws are on the underside of the Packet Engine and screw into standoffs on the Packet Accelerator.

(5) Gently pull up on the Packet Accelerator to free the pin header from the pins on the Packet Engine. If the Packet Accelerator does not lift freely, gently rock the rear of the Packet Accelerator from side to side to loosen it from the pins.

(6) Place the Packet Accelerator in its protective packaging.

(7) Reinstall the Packet Engine. (See Section 5.2.1.)

(8) Reinstall the NIM(s) or UMM you removed from the middle and top slots. (See Section 5.11.1 for instructions on installing a NIM or Section 5.8.1 on installing the UMM.)

5.7 INSTALLING A DRAM UPGRADE

The following procedure describes how to upgrade the DRAM on the Packet Engine.

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–7 shows the location of the DRAM socket. Refer to this figure as you perform the following procedure.

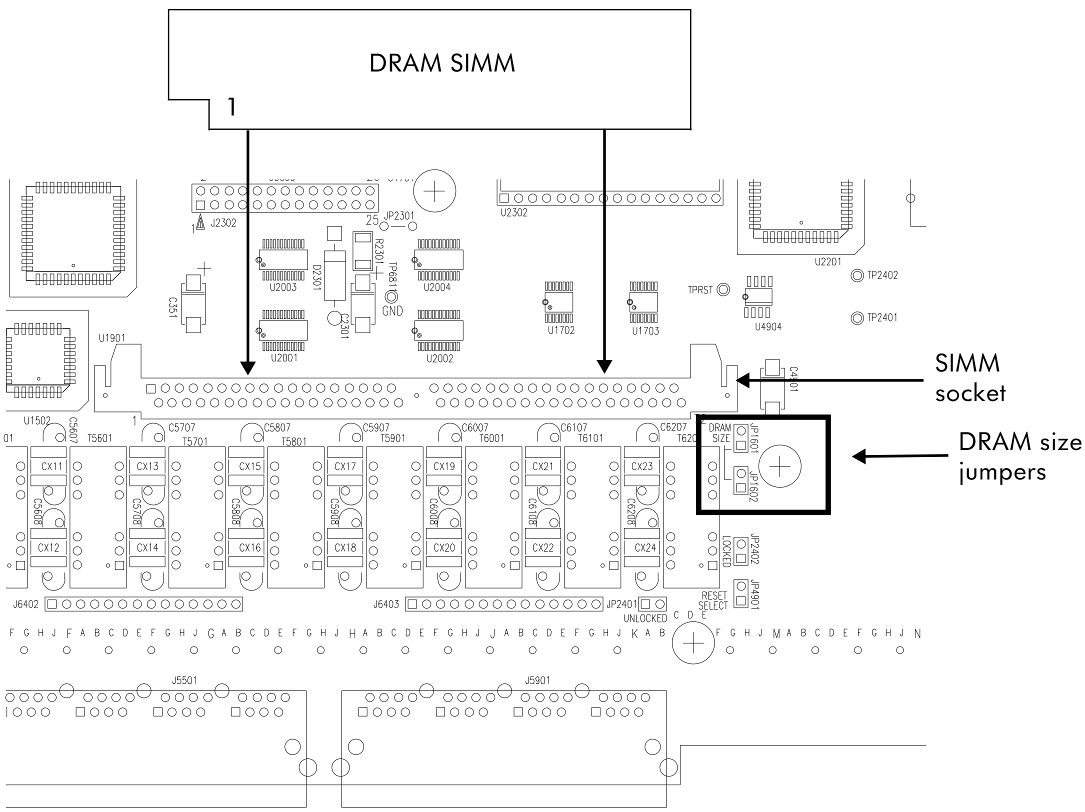


FIGURE 5–7 DRAM socket and DRAM size jumpers.

To install a DRAM upgrade:

(1) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Remove the NIM or UMM from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) Remove the Packet Engine. (See Section 5.2.2.)

(4) Press the two plastic tabs that secure the old DRAM SIMM to release it from the socket.

(5) Holding the old DRAM SIMM by its edges, gently remove it and place it in a protective package.

(6) Holding the new DRAM SIMM by its upper edges, align it over the SIMM socket. The DRAM SIMM is properly aligned when the copper-colored edge is facing down and the number “1” appears in the left corner, to the left of the copper-colored edge.

(7) Gently press down and back on the DRAM SIMM to seat it in the socket. The DRAM SIMM will snap into place when fully seated.

(8) Remove the jumpers from the DRAM SIZE jumper pins. These jumper pins are located by the lower right corner of the DRAM socket (see Figure 5–7 on page 88).

(9) Set the jumpers according to the size of the DRAM SIMM you are installing:

- For a 1 or 2 MB DRAM SIMM, set both jumpers out. (To avoid losing the jumpers, you can place them over just one pin on the jumper headers.)
- For a 4 or 8 MB DRAM SIMM, set JP1601 out and JP1602 in.
- For a 16 or 32 MB DRAM SIMM, set JP1601 in and JP1602 out.

(10) Reinstall the Packet Engine. (See Section 5.2.1.)

(11) Reinstall the NIM(s) or UMM you removed from the middle and top slots. (See Section 5.11.1 for instructions about installing a NIM or Section 5.8.1 on installing the UMM.)

(12) After you have completely finished the hardware changes, power on the system and issue the **`mgmt showcfg`** command to verify the DRAM amount. (See Section 9.3.6 on page 160.)

5.8 UMM (UNIVERSAL MEDIA MODULE)

The following procedures describe how to install and remove the UMM.

5.8.1 Installing the UMM

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

To install the UMM:

(1) If the UMM does not already contain the EMAs or AUI Media Cables you need, use the appropriate procedures in this chapter to install them:

- To install an EMA, use the procedure in Section 5.9.2.
- To install an AUI Media Cable, use the procedure in Section 5.9.4.

(2) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(3) Use the flat-head screwdriver to remove the NIM or coverplate from the middle slot in the chassis, as applicable.

(4) Remove the Packet Engine. (See Section 5.2.2.)

(5) Attach the ribbon cables to the UMM headers on the Packet Engine. The cables are identical, as are the headers on the cables. You can attach either cable to either header. Also, you can attach either end of a cable to either the UMM or the Packet Engine. However, the connectors are keyed and can go into the headers only one way.

Figure 5–8 shows the location of the UMM headers on the Packet Engine.

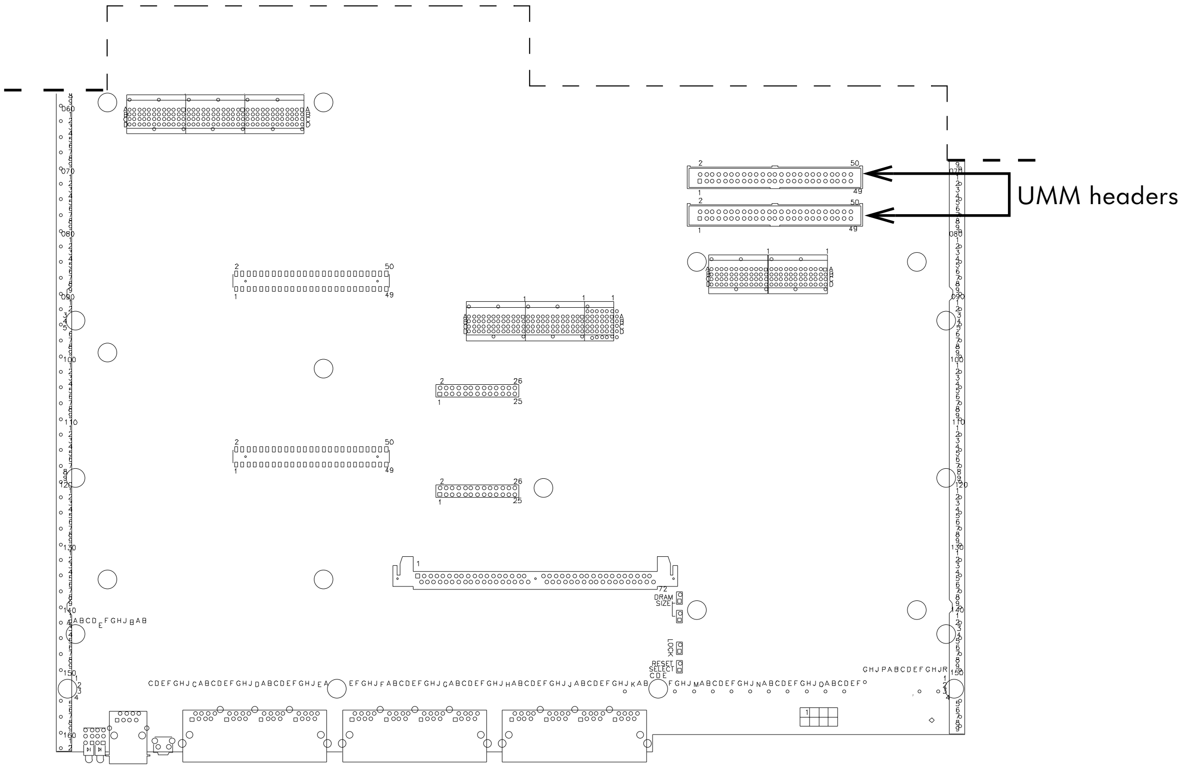


FIGURE 5–8 Location of UMM headers on Packet Engine.

(6) Connect the other ends of the ribbon cables to the headers on the UMM. Make sure the cable on the Packet Engine UMM header closest to the front of the chassis is attached to the rear header on the UMM. (The cables should not twist or wrap.)

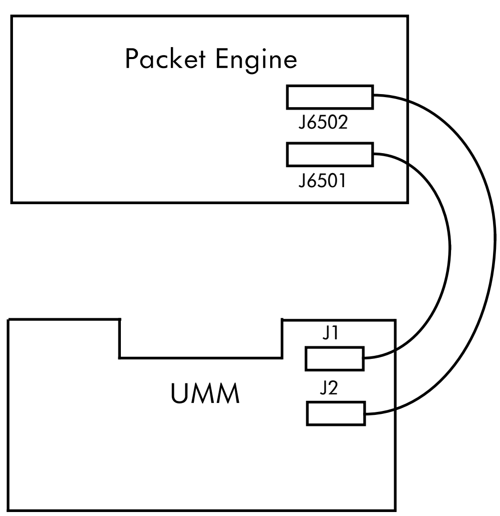


FIGURE 5–9 Connectors on the UMM and the Packet Engine.

(7) Carefully place the UMM on top of the Packet Engine, making sure the UMM rests on both the support bars on the sides of the Packet Engine. Align the front panels of the UMM and Packet Engine so they are flush.

(8) Holding the UMM and Packet Engine so that the UMM cannot slip off of the Packet Engine, slide the two modules into the chassis.

(9) Use the flat-head screwdriver to tighten the screws on the UMM and Packet Engine front panels.

(10) If the Packet Engine contains the Packet Channel backplane, use the #2 Phillips-head screwdriver to tighten the thumbscrews that secure the backplane to the chassis.

(11) Reinstall the NIM or coverplate you removed from the top slot. (See Section 5.11.1 for instructions about installing a NIM.)

5.8.2 Removing the UMM

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–8 on page 91 shows where the headers for the UMM ribbon cables are located. Refer to this figure as you perform the following procedure.

To remove the UMM:

(1) If the top slot in the chassis contains a NIM or cover plate, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Use the flat-head screwdriver to loosen the screws on the UMM and Packet Engine front panels.

(3) If the Packet Engine contains the Packet Channel backplane, use the #2 Phillips-head screwdriver to loosen the thumbscrews that secure the backplane to the chassis.

(4) Carefully pull the UMM and Packet Engine out of the chassis. Make sure you hold both modules firmly, so the UMM does not slip off of the Packet Engine.

(5) Place the UMM and Packet Engine on the work surface you have prepared.

(6) Disconnect the ribbon cables from the UMM and remove the UMM from off of the Packet Engine.

(7) Unplug the UMM ribbon cables from the Packet Engine and store them with the UMM in a safe place.

(8) Insert the Packet Engine back into the chassis, then use the flat-head screwdriver to tighten the screws on the Packet Engine front panel.

(9) If the Packet Engine contains the Packet Channel backplane, use the #2 Phillips-head screwdriver to tighten the thumbscrews that secure the backplane to the chassis.

(10) If you plan to install a NIM in place of the UMM, use the procedure in Section 5.11.1. Use this same procedure to install a NIM in the top slot, if applicable.

(11) If you plan to leave the UMM slot empty, use the flat-head screwdriver to install a coverplate (part# 171-1207-0001) over the slot.

5.9 EMA OR AUI MEDIA CABLE

The following sections describe how to remove and install PowerHub 6000 EMAs and the AUI Media Cable on the UMM.

5.9.1 Removing an EMA

This procedure applies to the following types of PowerHub 6000 EMAs:

- 10Base-FB.
- 10Base-FL.
- BNC.

However, this procedure does not apply to AUI Media Cables. To remove an AUI Media Cable, use the procedure in Section 5.9.3.

For this procedure, you need the following:

- A #1 Phillips-head screwdriver.
- A regular flat-head screwdriver.
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–10 shows how an EMA is removed from the UMM.

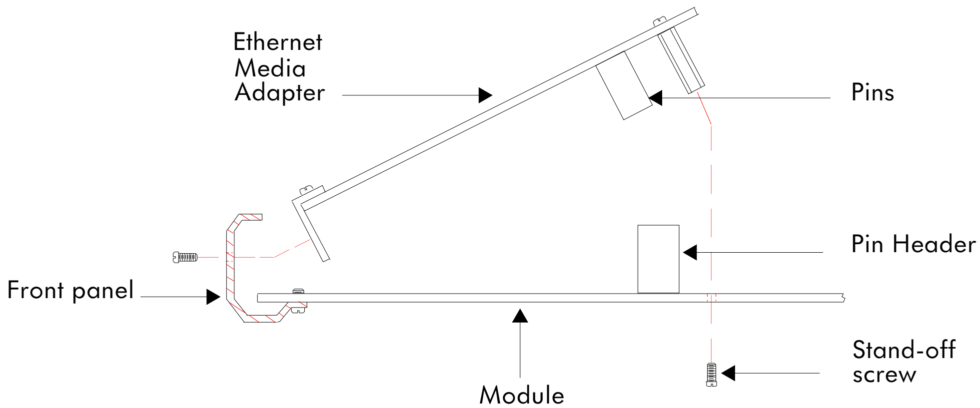


FIGURE 5–10 How an EMA is removed from the UMM.

To remove an EMA from the UMM:

(1) Use the procedure in Section 5.8.2 to remove the UMM from the chassis.

(2) Use the #1 Phillips-head screwdriver to remove the 4-40 1/4” pan-head screw that secures the EMA to the UMM. This screw is located on the underside of the UMM. Do not remove the screw on the EMA itself. This screw holds the standoff onto the EMA.

(3) Use the #1 Phillips-head screwdriver to remove the two 4-40 3/8” pan-head screws that secure the EMA to the UMM front panel.

(4) Grab the pin-end of the EMA, making sure not to grab the pin connections. (These can become bent or poke holes in your fingers.)

(5) Gently pull up on the pin-end of the EMA to loosen the pins from the socket on the UMM. If the pins do not come loose, gently rock the rear of the EMA from side to side while pulling up. Stop pulling as soon as the pins come loose! If you continue to pull, you might crack the EMA.

(6) When the EMA pins are free from the socket on the UMM, gently pull the EMA straight back out of the UMM front panel. If the EMA gets caught, gently jiggle it free. Do not force the EMA to come free; you might break components.

(7) When the EMA is completely free of the UMM, either place it in its protective container for storage or, if you plan to install it in another UMM, place it on a grounded table.

(8) If you do not plan to immediately replace the EMA with another EMA or with an AUI Media Cable, install a coverplate (part# 171-1230-0001) over the unused EMA position.

NOTE: Make sure you install the appropriate cover plates over all unused slots, EMA positions, and power supply bays.

(9) If you do not plan to remove any more EMAs or AUI Media Cables, use the procedure in Section 5.8.1 to reinstall the UMM.

(10) Reboot the system to activate the changed system configuration.

5.9.2 Installing an EMA

This procedure applies to the following types of PowerHub 6000 EMAs:

- 10Base-FB.
- 10Base-FL.
- BNC.

However, this procedure does not apply to AUI Media Cables. To install an AUI Media Cable, see Section 5.9.4.

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #1 Phillips-head screwdriver.
- A #2 Phillips-head screwdriver (if the Packet Engine contains the Packet Channel backplane).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–10 shows how an EMA is attached to a module. You might want to refer to this figure as you perform the following procedure.

To install the EMA:

(1) If you have not already done so, use the procedure in Section 5.8.2 to remove the UMM from the chassis.

(2) Hold the EMA by the pin-end. Make sure you do not touch the pins; they can become bent or poke holes in your finger.

(3) Gently insert the segment connector on the front of the EMA through the appropriate hole on the UMM front panel. Ensure that the segment connector is fully forward by pushing on it until it is firmly in place.

(4) Align the EMA pins directly over the socket on the UMM.

(5) When the pins are aligned over the socket, firmly press the pins down into the socket. Make sure the pins go all the way into the socket.

(6) Secure the EMA to the UMM. Use the #1 Phillips-head screwdriver to insert the 4-40 1/4” pan-head screw into the bottom of the EMA, making sure the screw is fully tightened. The screw hole is located on the underside of the UMM.

(7) Use the #1 Phillips-head screwdriver to insert the two 4-40 3/8” pan-head screws into the UMM front panel.

(8) If you do not plan to install any more EMAs or AUI Media Cables, use the procedure in Section 5.8.1 to reinstall the UMM.

(9) Reboot the system to activate the changed system configuration.

5.9.3 Removing an AUI Media Cable

The procedure for removing an AUI Media Cable differs from the procedure for removing an EMA. This procedure applies only to the AUI Media Cable. To remove an EMA, use the procedure in Section 5.9.1.

For this procedure, you need the following:

- A small flat-head screwdriver.
- A regular flat-head screwdriver.
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–11 shows how an AUI Media Cable is attached to the front panel of the UMM. You might want to refer to this figure as you perform the following procedure.

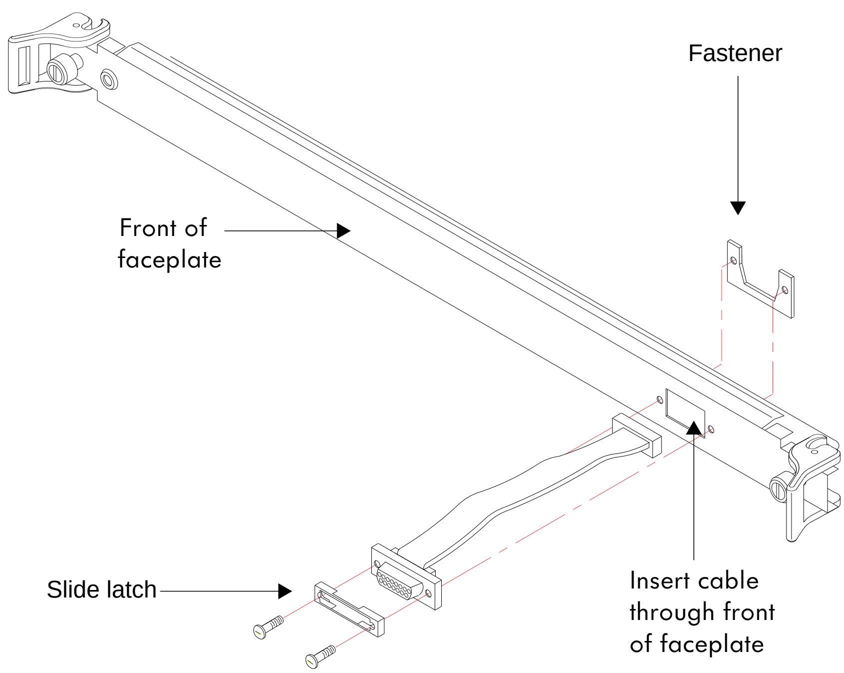


FIGURE 5–11 How an AUI Media Cable is attached to the UMM.

To remove an AUI Media Cable:

(1) If you have not already done so, use the procedure in Section 5.8.2 to remove the UMM from the PowerHub chassis.

(2) Use the small flat-head screwdriver to remove the two 4-40 3/8” screws that fasten the AUI connector and slide latch to the UMM front panel. When you remove the screws, the slide latch and the fastener that secures the connection from the rear of the UMM front panel (see Figure 5–11) are freed.

(3) Collect the two screws, the slide latch, and the fastener and place them in a container for storage. You might need them if you choose to reinstall the AUI Media Cable in this UMM or another UMM at a later time.

(4) Pull the AUI Media Cable’s pin connector loose from the socket on the UMM. If the connector does not come off easily, gently rock the connector from side to side to free it from the pins.

(5) From the rear, pull the connector out of the UMM front panel.

(6) Unless you plan to immediately reinstall the AUI Media Cable on the same or another UMM, place the removed AUI Media Cable in an appropriate storage container. You might want to store the screws, slide latch, and fastener in the same container.

(7) If you plan to install an EMA in place of the removed AUI Media Cable, use the procedure in Section 5.9.2. If you plan to leave the segment position unoccupied, install a cover plate (part# 171-1230-0001).

NOTE: Make sure you install the appropriate cover plates over all unused openings.

(8) Reboot the system to activate the changed system configuration.

5.9.4 Installing an AUI Media Cable

The procedure for installing an AUI Media Cable differs from the procedure for installing an EMA. This procedure applies only to the AUI Media Cable. To install an EMA, use the procedure in Section 5.9.2.

For this procedure, you need:

- A regular flat-head screwdriver.
- A small flat-head screwdriver.
- Two 4-40 3/8” screws (supplied with the AUI Media Cable).
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–11 on page 96 shows how an AUI Media Cable is attached to the UMM front panel. You might want to refer to this figure as you perform the following procedure.

To install an AUI Media Cable:

(1) If you have not already done so, use the procedure in Section 5.8.2 to remove the UMM from the PowerHub chassis.

(2) If the position in which you plan to install the AUI Media Cable contains an EMA, use the procedure in Section 5.9.1 to remove the EMA.

(3) From the rear of the UMM front panel, insert the segment connector on the AUI Media Cable into the front panel.

(4) Align the connector pins at the rear of the cable directly over the header on the UMM. The AUI header is the one closest to the rear of the UMM.

(5) When the connector pins are aligned, gently press them all the way down into the UMM header.

(6) Insert the fastener directly behind the segment connector on the rear of the front panel. The two screw holes in the fastener must line up with the screw holes in the front panel.

(7) Use the small flat-head screwdriver to start (but do not screw all the way in) the supplied 4-40 3/8” screws in the front of the AUI connector. The screws must screw into the fastener behind the UMM front panel.

(8) Place the slide latch over the screws and onto the segment connector. Make sure the round screw holes on the slide latch face outward, toward you.

(9) Use the small flat-head screwdriver to fasten the screws firmly into place, securing the segment connector firmly to the face plate. Do not over-tighten the screws; the slide latch must be able to switch left and right.

(10) Repeat this procedure for each AUI Media Cable. If you plan to install an EMA in place of the removed AUI Media Cable, use the procedure in Section 5.9.2. When you are finished, use the procedure in Section 5.8.1 to reinstall the UMM.

NOTE: Make sure you install the appropriate cover plates over all unused openings.

(11) Reboot the system to activate the changed system configuration.

5.10 PACKET CHANNEL BACKPLANE

The following procedures describe how to install and remove the Packet Channel backplane.

5.10.1 Installing the Packet Channel Backplane

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #2 Phillips-head screwdriver.
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–12 shows where the Packet Channel backplane is installed on the Packet Engine. Refer to this figure as you perform the following procedure.

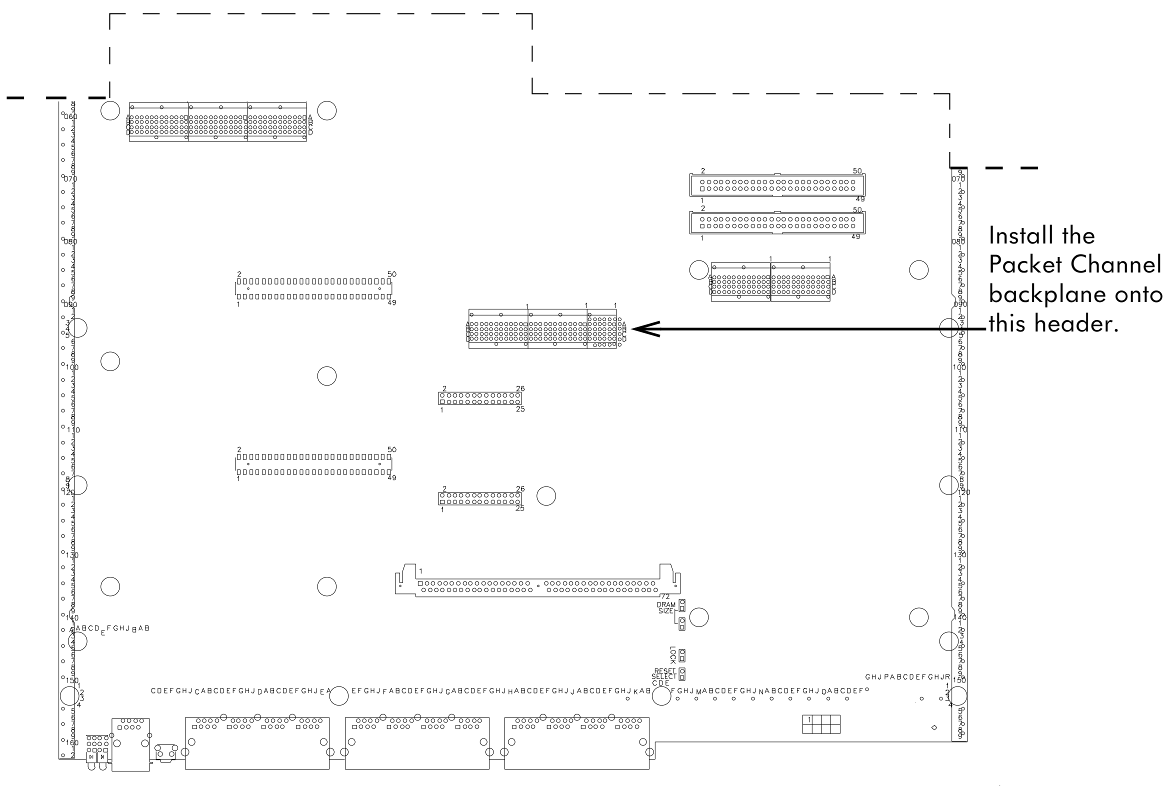


FIGURE 5–12 Where the Packet Channel backplane is attached to the Packet Engine.

To install the Packet Channel backplane:

(1) Use the flat-head screwdriver to remove the coverplate from the top slot in the chassis.

(2) Use the flat-head screwdriver to remove the coverplate or UMM from the middle slot in the chassis. (To remove the UMM, use the procedure in Section 5.8.2.)

(3) Remove the Packet Engine. (See Section 5.2.2.)

(4) Grab the backplane by the top corners, making sure the screws are facing the front of the chassis and the pin header on the edge of the backplane is facing down, toward the Packet Engine.

(5) Align the pin header over the corresponding pins on the Packet Engine. The backplane is properly aligned when the left side of the header is flush with the left side of the pin socket on the Packet Engine.

(6) Gently press down on the backplane to seat the header on the pins.

(7) When the backplane is completely seated on the Packet Engine, slide the Packet Engine (with the backplane attached) back into the chassis. Two guide pins on the chassis fit into corresponding holes on the edges of the backplane. Make sure the guide pins go through the holes in the backplane.

(8) When the Packet Engine is all the way in the chassis, tighten the two thumbscrews that secure the backplane to the support bar in the chassis.

(9) Install a NIM, UMM, or coverplate (part# 171-1207-0001) in the middle slot. (See Section 5.11.1 for instructions on installing a NIM or Section 5.8.1 on installing a UMM.)

(10) Install a NIM or coverplate in the top slot. (See Section 5.11.1 for instructions on installing a NIM.)

NOTE: Make sure you install the appropriate cover plates over all unused openings.

5.10.2 Removing the Packet Channel Backplane

For this procedure, you need the following:

- A regular flat-head screwdriver.
- A #2 Phillips-head screwdriver.
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

Figure 5–12 on page 99 shows where the Packet Channel backplane is installed on the Packet Engine. Refer to this figure as you perform the following procedure.

To remove the Packet Channel backplane:

(1) Remove the NIM or coverplate from the top slot in the chassis. To remove a NIM, use the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot, before attempting to remove the Packet Engine. Do not attempt to remove a module until all modules above it have been removed.

(2) Remove the NIM or UMM from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) Remove the Packet Engine (with the backplane attached). Don't forget to loosen the two thumbscrews that secure the Packet Channel backplane to the chassis. (See Section 5.2.2.)

(4) Grab each upper corner of the backplane and pull the backplane off of the pins on the Packet Engine. If the backplane does not come free, gently move it from side to side to loosen it from the pins on the backplane.

(5) Store the backplane in a protective container.

(6) If you plan to install another backplane, use the procedure in Section 5.10.1 on page 99 to do so.

(7) Reinstall the Packet Engine. (See Section 5.2.1.)

(8) Reinstall the coverplates, UMM, or NIMs you removed from the middle and top slots. (See Section 5.11.1 for instructions on installing a NIM or Section 5.8.1 on installing a UMM.)

> **NOTE**: Make sure you install the appropriate cover plates over all unused openings.

5.11 NIM (NETWORK INTERFACE MODULE)

The following procedures describe how to install and remove a NIM.

> **NOTE**: NIMs require the Packet Channel backplane. Also, you cannot install a NIM in the top slot if the middle slot is empty.

5.11.1 Installing a NIM

For this procedure, you need the following:

- A regular flat-head screwdriver.
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

To install a NIM:

(1) If the top slot in the chassis contains a NIM, remove it using the procedure in Section 5.11.2.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot. Do not attempt to remove a module until all modules above it have been removed.

(2) If you plan to install the NIM in the middle slot, remove the NIM, UMM, or coverplate from the middle slot in the chassis, as applicable:

- To remove a NIM, use the procedure in Section 5.11.2.
- To remove the UMM, use the procedure in Section 5.8.2.

(3) Align the rear corners of the NIM evenly in the slot, then slide the NIM into place. The NIM is inserted properly when the rear of the NIM fits into the corresponding notches on the Packet Channel backplane.

(4) Use the flat-head screwdriver to tighten the screws on the NIM faceplate.

(5) Reinstall the module(s) or coverplate you removed, as applicable.

> **NOTE**: Make sure you install the appropriate cover plates over all unused openings.

5.11.2 Removing a NIM

For this procedure, you need the following:

- A regular flat-head screwdriver.
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

To remove a NIM:

(1) If the chassis contains a NIM in the top slot, but you plan to remove a NIM from the middle slot, first perform this procedure for the NIM in the top slot.

> **IMPORTANT**: You must remove the modules from the top down. That is, remove the module in the top slot, then remove the module in the middle slot. Do not attempt to remove a module until all modules above it have been removed.

(2) Use the flat-head screwdriver to loosen the screws that secure the NIM to the front of the chassis.

(3) Pull the NIM out of the chassis.

(4) If you do not plan to reinstall the NIM, store it in protective packaging.

(5) If you do not plan to install another module in place of the removed NIM, install a coverplate (part# 171-1207-0001) over the slot. If you do plan to install another module, use the appropriate procedure to do so:

- If you plan to install the UMM, see Section 5.8.1.
- If you plan to install a NIM, see Section 5.11.1.

5.12 CHANGING THE LOCK SWITCH JUMPER SETTING

You can change the setting of the Lock Switch jumper to disable the Lock Switch on the front of the Packet Engine, permanently setting the Lock Switch on or off. When the Lock Switch is on, the appropriate root or monitor password must be entered before a command prompt is displayed. When the Lock Switch is off, no password is required, and anyone can access the PowerHub user interface by opening a TTY or TELNET session to the hub.

> **CAUTIONS**: Static electricity can damage the electronic components of the Packet Engine. Make sure you take appropriate precautions as described in Section 4.1.1 on page 50.
>
> Do not touch the electronic components of the Packet Engine directly.

5.12.1 Permanently Locking the Switch

For this procedure, you need the following:

- A regular flat-head screwdriver.
- An ESD wrist-strap.
- If you plan to remove components, you also need a grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

To force the Lock Switch to always be locked:

(1) Use the procedure in Section 5.2.2 to remove the Packet Engine. When you remove it, place it on a grounded table or other work surface.

> **CAUTION:** Never attempt to add, remove, or modify components on a Packet Engine or NIM when the module is still in the chassis. Components can become broken or damaged through electrostatic discharge, or the module itself can become cracked through improper handling.

(2) Locate the Lock Switch headers and jumper. The Lock Switch headers and jumper are located on the right edge of the Packet Engine, about half-way toward the rear of the module. Figure 5–13 shows the Lock Switch headers in detail.

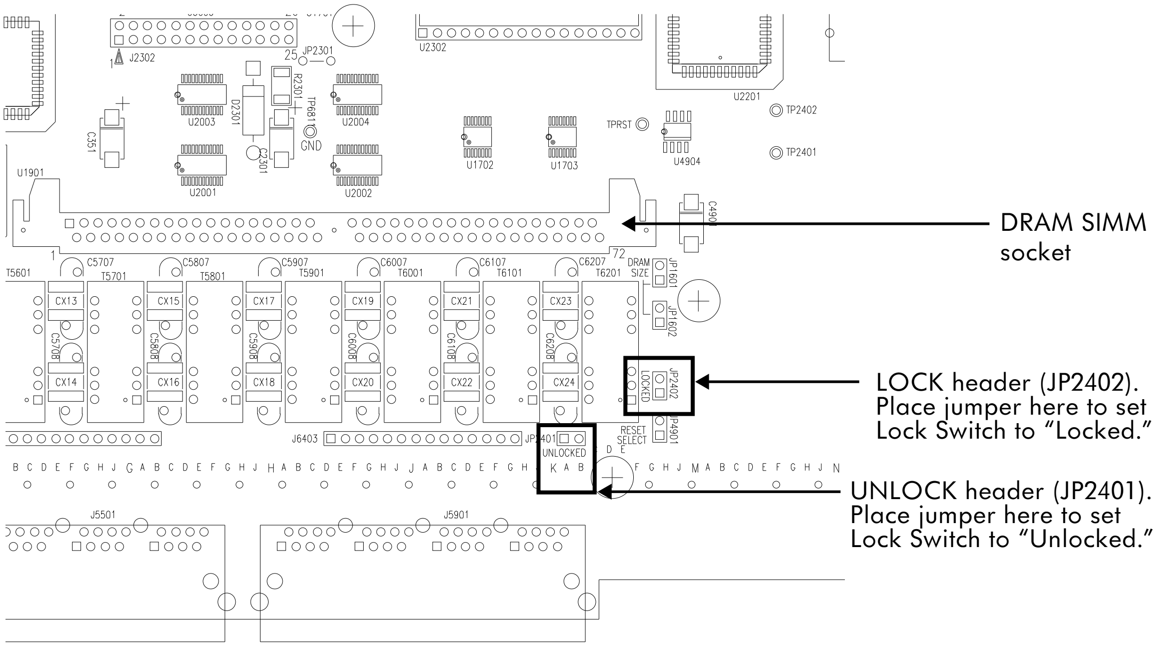


FIGURE 5–13 Location of the Lock Switch jumpers.

(3) After locating the pins and jumper, remove the jumper from its current position. Make sure you do not touch other components on the Packet Engine.

(4) Place the jumper over both the LOCK pins. Make sure the jumper is seated all the way onto the pins.

(5) Use the procedure in Section 5.2.1 on page 74 to reinstall the Packet Engine.

5.12.2 Permanently Unlocking the Switch

To force the Lock Switch to always be unlocked:

(1) Use the procedure in Section 5.2.2 on page 76 to remove the Packet Engine.

> **CAUTION:** Never attempt to add, remove, or modify components on a Packet Engine or NIM when the module is still in the chassis. Components can become broken or damaged through electrostatic discharge, or the module itself can become cracked through improper handling.

(2) Locate the Lock Switch headers and jumper. The Lock Switch headers and jumper are located on the right edge of the Packet Engine, about half-way toward the rear of the module. Figure 5.12 shows the Lock Switch headers in detail.

(3) When you have located the pins and jumper, remove the jumper from its current position. Make sure you do not touch other components on the Packet Engine.

(4) Place the jumper over the two UNLOCK pins. Make sure the jumper is seated all the way onto pins.

(5) Use the procedure in Section 5.2.1 on page 74 to reinstall the Packet Engine.

5.12.3 Restoring Control to the Lock Switch

To give control to the Lock Switch on the front panel of the Packet Engine:

(1) Use the procedure in Section 5.2.2 on page 76 to remove the Packet Engine. When you remove it, place it on a grounded table or other work surface.

> **CAUTION:** Never attempt to add, remove, or modify components on a Packet Engine or NIM when the module is still in the chassis. Components can become broken or damaged through electrostatic discharge, or the module itself can become cracked through improper handling.

(2) Locate the Lock Switch headers and jumper. The Lock Switch headers and jumper are located on the right edge of the Packet Engine, about half-way toward the rear of the module. Figure 5.12 shows the Lock Switch headers in detail.

(3) When you have located the pins and jumper, remove the jumper from its current position. Make sure you do not touch other components on the Packet Engine.

(4) Place the jumper over just one pin (but not both pins) on the Lock or Unlock pins for storage.

5.13 REMOVING AND RE-INSTALLING THE CHASSIS COVER

You can perform all of the procedures in this chapter without removing the chassis' top cover. However, if you would prefer to perform a a particular procedure with the cover removed from the chassis, use the following procedure.

> **CAUTIONS**: Electrostatic discharges (shocks) can permanently damage the PowerHub 6000's electronic components. Carefully read all of Section 4.1 on page 49, "Safety Precautions," before you remove the chassis cover.
>
> Also, to prevent personal injury or damage to the PowerHub components, always turn off the Power Supplies and disconnect the power cables before performing any of the procedures in this chapter.

5.13.1 Removing the Chassis Cover

For this procedure, you need the following:

- A #2 Phillips-head screwdriver.
- A regular flat-head screwdriver (if the chassis is mounted in a rack).
- An ESD wrist-strap.
- A grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

(1) Turn the switches on both power supplies off.

(2) Remove the power cables first from the power supplies, then from the wall sockets.

(3) If the hub is installed in a rack, make a note of the segment cables and TTY cable attached to the chassis, then remove the cables.

(4) If the hub is installed in a rack, remove the four screws that secure the mounting brackets to the rack.

> **CAUTION**: When you loosen these screws, the chassis will be loosened from the rack. Make sure you or an assistant holds the chassis when you loosen the screws.

(5) Place the chassis on the grounded work surface you have prepared.

(6) Use the #2 Phillips-head screwdriver to remove the 3/8" screws that hold the mounting brackets to the chassis. Place these screws and the mounting brackets aside. Do not mix other chassis screws with these screws. If the chassis does not have mounting brackets, you still need to remove the four 1/4" screws. These screws are located along the frontmost edges of the left and right sides of the chassis.

(7) Use the #2 Phillips-head screwdriver to remove the remaining fourteen screws holding the top cover onto the chassis. The screws are located on the left and right sides of the chassis, near the top. Do not mix these screws with the four screws that held the mounting brackets to the chassis.

(8) Lift the top cover off of the chassis. If the cover does not lift easily, check to make sure you removed all of the screws.

5.13.2 Re-Installing the Chassis Cover

For this procedure, you need the following:

- A #2 Phillips-head screwdriver.
- A regular flat-head screwdriver (if the chassis is to be mounted in a rack).
- An ESD wrist-strap.
- A grounded work surface, such as a grounded metal table or a table covered with a grounded, rubberized mat. (See Section 4.1 on page 49.)

(1) Check the open chassis to make sure no components (daughter cards, Flash Memory Module) are loose. Also check to make sure you have no loose screws in the chassis.

(2) Place the top cover onto the chassis, aligning the screw holes in the cover with corresponding holes in the chassis.

(3) If the chassis is to be mounted in a rack, align a mounting bracket over the two corresponding screw holes in the chassis.

(4) Use the #2 Phillips-head screwdriver to insert the 3/8" screws that hold the mounting bracket to the chassis. Do not attempt to use 1/4" screws to secure the bracket to the chassis. Repeat Step 3 and this step for the other bracket.
(See Section 4.3.1 on page 53 for information on selecting and installing mounting brackets.)

(5) Use the #2 Phillips-head screwdriver to insert the remaining 14 screws that hold the top cover onto the chassis. All these screws are 1/4". Do not use any of the screws that hold the mounting brackets to the chassis.

(6) Lift the chassis into place in the rack, then use the flat-head screwdriver to insert the four 1/2" pan-head screws that secure the mounting brackets to the rack.

> **CAUTION**: Make sure you or an assistant holds the chassis as you tighten the screws.

(7) Reinsert the TTY and segment cables.

(8) Plug the power cables into the wall sockets, then into the power supplies.

(9) Turn on the power supplies.

6 Installing Software Upgrades

This chapter contains procedures for installing upgrades for the following types of software:

- Packet Engine boot PROM. (See Section 6.1 on page 110.)
- System software. (See Section 6.2 on page 115.)

> **CAUTIONS**: Before performing a software upgrade, always consult the Release Notes or Upgrade Notes that come with the software upgrade. The notes might contain information or procedural changes not anticipated in this chapter. Also, if you are upgrading the Packet Engine boot PROM, the upgrade diskette might contain an upgraded PROM Programming Utility.
>
> Disconnect your segment cables before performing any of the procedures in this chapter. Live network traffic can interfere with the upgrade.

We recommend that you always keep the software diskettes that contain the current and older versions of software as backups. If you ever need to re-install the current version or an older version of the software, you can use the diskettes.

Sometimes, the upgrade for one type of software requires an accompanying upgrade in other software types. For example, you might need to upgrade a boot PROM before you can install and use a system software upgrade.

Unless the documentation accompanying the software upgrade states otherwise, perform multiple upgrades in the following order:

(1) Packet Engine boot PROM.

(2) System and FDDI software.

REQUIREMENTS: To use the PowerHub ZMODEM commands to transfer files, your PC or Macintosh must have a terminal emulation program that supports the ZMODEM protocol. Otherwise, you must copy the upgrade files from the upgrade diskettes onto a TFTP server, then from the TFTP server to the PowerHub system as described in the following procedures.

6.1 UPGRADING THE PACKET ENGINE BOOT PROM

Upgrades for the Packet Engine boot PROM are shipped on 3-1/2” HD floppy diskettes that can be read by DOS-compatible machines. To install the upgrade, you must:

- Copy the upgrade files from the diskette onto a TFTP server, or onto a PC or Macintosh running ZMODEM. (The segment attaching the PowerHub system to the TFTP server must have an IP interface. To add an IP interface to the segment, use the **`ip add-interface`** command. See Section 4.3.4.4 on page 64 for instructions.)
- Download the upgrade files from the TFTP server, PC, or Macintosh onto the Flash Memory Module.
- Boot the PROM Programming Utility (contained on the Flash Memory Module).
- Install the upgrade.
- Reset the system under the new PROM.

The following procedure describes how to perform these steps.

CAUTION: Do not turn the hub off while the PROM is being upgraded. If you do, the hub will stop operating and you will not be able to boot the software. In the unlikely event of a power failure during the PROM upgrade, contact FORE Systems TAC for information on physically replacing the PROM.

NOTE: As described in Section 2.3.6 on page 27, the Flash Memory Module is an EEPROM. One of the characteristics of EEPROMs is that data can be read from them very rapidly, which makes them very useful for booting. However, reprogramming EEPROMs takes longer than reading information from them, and takes longer than writing information to a hard drive or floppy diskette. Consequently, you can expect this procedure to require from 25 to 40 minutes, depending upon the size and number of files you transfer.

To upgrade the Packet Engine boot PROM:

(1) Read the Release Notes or PROM Upgrade Instructions that came with the PROM upgrade. The notes will describe any changes to this procedure, and will tell you whether the floppy diskette includes an upgrade of the PROM Programming Utility (ppu-6pe). If the floppy diskette does contain an upgrade PROM Programming Utility, make sure you copy the utility along with the new boot PROM in this procedure.

(2) Connect the PC or Macintosh to the TTY1 port on the PowerHub system, if you have not already done so. For information about how to assemble the cable, see Section 4.3.2 on page 55.

(3) Copy the upgrade files (6pe.prm and ppu-6pe) from the floppy diskette onto a TFTP server or onto the hard drive of the PC or Macintosh. Unless specified otherwise in the software upgrade notes, you need to copy only these files.

(4) Copy the files from the TFTP server, PC, or Macintosh onto the hub's Flash Memory Module.

- If you copied the files onto a TFTP server, issue the following command at the hub's runtime command prompt:

  tftp get [-h *<host>***]** *<remfile>* *<locfile>*

  where:

  -h *<host>* Specifies the IP address of the TFTP server. Unless you have already specified a default TFTP server using the **tftp set** command, you need to include this argument. For information on the **tftp set** command, see the *PowerHub Software Manual, V 2.6* (Rev C).

  <remfile> Specifies the name of the remote file. Specify the name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called fore and this directory is specified as the TFTP home directory, do not specify fore as part of the file name.

> **NOTES**: Some TFTP servers require that the remote file name exist on the server before you can write to that file name. If your server requires that the file name already exist, create a short file (named the same as your configuration file) on the server, then specify that file name for *<remfile>*.
>
> Also, on some TFTP servers, including servers running Sun/OS 4.x, files that you overwrite on the server are not properly truncated. When you overwrite an existing file on the TFTP server, if the older version of the file is longer than the new file, the older version is not truncated properly by the server. As a result, the new version of the file contains part of the older version of the file.

  <locfile> Specifies the file name on the Flash Memory Module. Specify the names 6pe.prm and ppu-6pe.

- If you copied the file onto a PC or Macintosh that is running ZMODEM:

(a) Start the terminal emulation program on your PC or Macintosh, if you have not already done so. Make sure you select the ZMODEM protocol, set for a 32-bit CRC.

(b) Reboot the system. Watch the boot messages displayed on the management terminal until the following message is displayed:

`Hit any key to abort boot [<seconds>]`

where <*seconds*> counts down from five seconds. While the system is counting down the seconds, press any key. The `<PROM-6pe>` command prompt will be displayed on the management terminal.

(c) From the `<PROM-6pe>` prompt on the hub, issue the following command[1]: **`zreceive`**

> **NOTE**: Some terminal emulation programs transfer files to the PowerHub system faster than the EEPROM (Flash Memory Module) can accept the data. If this occurs, your terminal emulation application might display CRC error messages. You can safely disregard these messages. They indicate that the EEPROM is not able to accept more data until the data already received is written onto the EEPROM. The terminal emulation application will continue to send the remaining portion of the file until all of the file is written to the EEPROM.

(d) Use the terminal emulation application on the PC or Macintosh to use ZMODEM to send the `6pe.prm` and `ppu-6pe` files to the hub. The procedure for sending the files depends upon your application. See your ZMODEM documentation for details.

(5) At the `<PROM-6pe>` or runtime command prompt, enter the following command, then press Enter:

`nvram set locbdfile bootdef.ppu`

This command changes the boot definition file from `bootdef` (which loads the system software) to `bootdef.ppu`, which loads the PROM Programming Utility.

> **NOTE**: If the upgrade diskette contains an upgrade of the PROM Programming Utility (`ppu-6pe`), make sure you have copied it onto the Flash Memory Module, using the appropriate instructions in Step 4.

1. The PowerHub boot PROM contains commands in addition to **`zreceive`**, **`zsend`**, and the other commands documented in Chapter 10. The additional commands are used by FORE Systems TAC and can give unexpected results. Use the additional commands only if advised by FORE Systems TAC.

(6) Issue one of the following commands to reboot the system:

- From the `<PROM-6pe>` prompt, issue the **`boot fm`** command.
- From the runtime command prompt, issue the **`mgmt reboot`** command.

> **NOTE**: If the boot order is **`n`** or **`nm`** (netboot or netboot first), you must interrupt the boot process, when prompted, to access the `<PROM-6pe>` prompt. From the `<PROM-6pe>` prompt, issue the **`boot fm`** command to boot from the Flash Memory Module.

This causes the system to boot the PROM Programming Utility.

(7) When the utility is booted, a display similar to the following appears. The actual options and numbers might differ, depending upon the options available.

```
PROM Programming Utility Main Menu

     (1)  Display ethernet address
     (2)  Change ethernet address
     (3)  Reboot

     (4)  Program the 6pe MAIN PROM

Please enter the number of your selection:
```

(8) Select the option for programming the boot PROM (ex: "`Program the 6pe MAIN PROM`"). The following example shows the types of system messages you will see as the boot PROM is being upgraded.

```
Loading file "6pe.prm": ???? records, ???? bytes
Erasing...
Programming...
Verifying...
Programming operation complete
```

The steps for erasing and programming each take about five to ten seconds. The verification step takes less than one second.

(9) Reset the hub by entering the `Reboot` selection number at the following prompt and press enter:

```
Please enter the number of your selection:
```

Watch the boot messages displayed on the management terminal until the following message is displayed:

```
Hit any key to abort boot [<seconds>]
```

where <*seconds*> counts down from five seconds. While the system is counting down the seconds, press any key. The `<PROM-6pe>` command prompt will be displayed on the management terminal.

(10) At the `<PROM-6pe>` command prompt, enter the following command, then press Enter:

`nvram set locbdfile bootdef`

This command changes the boot definition file from `bootdef.ppu` back to `bootdef`, which loads the system software.

(11) Issue the following command, then press Enter:

`boot`

This resets the system under the new PROM and boots the system software.

(12) Verify that the Packet Engine boot PROM has been upgraded successfully by issuing the **`main version`** command. This command displays the currently installed versions of the system software and the Packet Engine boot PROM. For information about this command, see Section 8.3 on page 139.

6.2 UPGRADING THE SYSTEM SOFTWARE

To upgrade the system software, you copy the new image file (`6pe`) from the floppy diskette onto the hub's boot source(s). The following procedures describe how to do this.

Upgrades for the system software are shipped on 3-1/2" HD floppy diskettes that can be read by DOS-compatible machines. The procedure for upgrading the system software depends upon the boot source(s). Perform the applicable procedures for each boot source you plan to use for the hub, including any boot source you specify as the secondary source using the **`nvram set bo`** command (see Section 11.4 on page 214).

You also can use the following procedures to upgrade the FDDI software. Where a procedure specifies the file name `6pe`, substitute the file name `6fddi`.

6.2.1 Installing the Upgrade onto the Netboot (TFTP) Server

> **NOTE**: This procedure assumes that you have already configured the BOOTP server, TFTP server, client hub, and gateway hub (if applicable) for network booting. If you have not configured these devices for booting the software, see Section 4.3.4.2 on page 61, then return to this procedure.
>
> Also, the segment attaching the PowerHub system to the TFTP server must have an IP interface. To add an IP interface to the segment, use the **`ip add-interface`** command. (See Section 4.3.4.4 on page 64 for instructions.)

To install a system software upgrade onto your netboot TFTP server:

(1) Create a new subdirectory on the TFTP server for the new system software image file. If the netboot directories on your server follow the naming convention recommended in Section 4.3.4.2 on page 61, name the new directory as follows:

`fore/ph/images/`*`<version>`*

where *`<version>`* is the software version you are installing (ex: `6-2.6.3.0`, `6-2.6.3.1`, `6-2.6.3.2`, and so on).

(2) Copy the image file for the new system software (`6pe`) from the floppy diskette onto the TFTP server.

(3) Edit the pathname of the system software image file listed in the boot definition file on the TFTP server. For example, if the boot definition file contains the following line:

`fore/ph/images/6-2.6.3.0/6pe`

Change the line to the following:

`fore/ph/images/`*`<version>`*`/6pe`

where *`<version>`* is the software version you are installing (ex: `6-2.6.3.0`, `6-2.6.3.1`, `6-2.6.3.2`, and so on).

(4) Reset the hub. Provided the procedure is successful and your primary boot source is the network, the system boots the new system software.

(5) Verify that the system software has been upgraded successfully by issuing the **`main version`** command. This command displays the currently installed versions of the system software and the Packet Engine boot PROM. For information about this command, see Section 8.3 on page 139.

6.2.2 Installing the Upgrade onto the Flash Memory Module

To install the system software upgrade onto the Flash Memory Module, you must copy the image file (`6pe`) from the floppy diskette onto another device (TFTP server, PC, Macintosh, and so on), then copy it from that device onto the Flash Memory Module. Use one of the following procedures:

- If the device is a TFTP server, use the procedure in Section 6.2.2.1.
- If the device is a PC or Macintosh that supports ZMODEM, use the procedure in Section 6.2.2.2 on page 117.

6.2.2.1 TFTP Server

> **NOTE**: The segment attaching the PowerHub system to the TFTP server must have an IP interface. To add an IP interface to the segment, use the **`ip add-interface`** command. (See Section 4.3.4.4 on page 64 for instructions.)

To install a system software upgrade onto the Flash Memory Module using a TFTP server:

(1) Copy the image file for the new system software from the floppy diskette onto the TFTP server.

(2) Copy the image file for the new system software from the TFTP server onto the Flash Memory Module by issuing the following command at the hub's runtime command prompt:

`tftp get [-h` *`<host>`* **`]`** *`<remfile>`* *`<locfile>`*

where:

`-h` *`<host>`*	Specifies the IP address of the TFTP server. Unless you have already specified a default TFTP server using the **`tftp set`** command, you need to include this argument. For information on the **`tftp set`** command, see Chapter 4 in the *PowerHub Software Manual, V 2.6* (Rev C).
`<remfile>`	Specifies the name of the remote file. Specify the name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `fore` and this directory is specified as the TFTP home directory, do not specify `fore` as part of the file name.

> **NOTES**: Some TFTP servers require that the remote file name exist on the server before you can write to that file name. If your server requires that the file name already exist, create a short file (named the same as your configuration file) on the server, then specify that file name for *`<remfile>`*.
>
> Also, on some TFTP servers, including servers running Sun/OS 4.x, files that you overwrite on the server are not properly truncated. When you overwrite an existing file on the TFTP server, if the older version of the file is longer than the new file, the older version is not truncated properly by the server. As a result, the new version of the file contains part of the older version of the file.

`<locfile>` Specifies the file name on the Flash Memory Module. Specify the name `6pe`.

When you complete this step, the image file for the new system software should be present on the Flash Memory Module, with the file name `6pe`. To verify this, issue the following command: **`mgmt listdir`** (or **`mgmt dir`**).

(3) After the file is successfully transferred, reset the hub. You can do this by pressing the reset switch (RST) on the Packet Engine or by issuing the **`mgmt reboot`** command. Provided the procedure is successful, the system boots the new system software.

(4) Verify that the system software has been upgraded successfully by issuing the **`main version`** command. This command displays the currently installed versions of the system software and the Packet Engine boot PROM. For information about this command, see Section 8.3 on page 139.

6.2.2.2 ZMODEM

Reprogramming EEPROMs takes longer than reading information from them, and takes longer than writing information to a hard drive floppy diskette. Consequently, you can expect this procedure to require from 25 to 40 minutes, depending upon the size and number of files you transfer.

To install a system software upgrade onto the Flash Memory Module using a PC or Macintosh that supports ZMODEM:

(1) Connect the PC or Macintosh to the `TTY1` port on the PowerHub system, if you have not already done so. For information about how to assemble the cable, see Section 4.3.2 on page 55.

(2) Copy the new image file (`6pe`) for the system software from the floppy diskette onto the hard drive on the PC or Macintosh. Unless specified otherwise in the software upgrade notes, you need to copy only this file.

(3) Start the terminal emulation program on your PC or Macintosh, if you have not already done so.

(4) Reboot the hub. Watch the boot messages displayed on the management terminal until the following message is displayed:

`Hit any key to abort boot [<seconds>]`

where <*seconds*> counts down from five seconds. While the system is counting down the seconds, press any key. The `<PROM-6pe>` command prompt will be displayed on the management terminal.

(5) From the `<PROM-6pe>` prompt on the hub, issue the following command: **`zreceive`**

(6) In the terminal emulation application on the PC or Macintosh, select the `6pe` file and start the file transfer. The procedure for sending the file depends upon your ZMODEM application. See your ZMODEM documentation for details. Make sure you select the ZMODEM protocol, set for a 32-bit CRC.

> **NOTE**: Some terminal emulation programs transfer files to the PowerHub system faster than the EEPROM (Flash Memory Module) can accept the data. If this occurs, your terminal emulation application might display CRC error messages. You can safely disregard these messages. They indicate that the EEPROM is not able to accept more data until the data already received is written onto the EEPROM. The terminal emulation application will continue to send the remaining portion of the file until all of the file is written to the EEPROM.

When you complete this step, the image file for the new system software should be present on the Flash Memory Module, with the file name `6pe`. To verify this, issue the following command at the `<PROM-6pe>` prompt: **`ls`**

(7) After the file is successfully transferred, reset the hub. You can do this by pressing the reset switch (RST) on the Packet Engine or by issuing the **`boot fm`** command. Provided the procedure is successful, the system boots the new system software.

(8) Verify that the system software has been upgraded successfully by issuing the **`main version`** command. This command displays the currently installed versions of the system software and the Packet Engine boot PROM. For information about this command, see Section 8.3 on page 139.

Part 3: Configuration

This part describes how to issue commands, then describes the commands you use to configure the PowerHub 6000.

NOTE: The commands described in this part do not configure the hub for bridging or routing:

- For information on configuring the hub for bridging and IP routing, see the *PowerHub Software Manual, V 2.6* (Rev C).
- For information on configuring the hub for AppleTalk, DECnet, IPX, or IP Security, see the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).

This part contains the following chapters:

Chapter 7: Getting Started with the User Interface

Describes the user interface, and how to issue commands, access help, use the command history, and define command aliases.

Chapter 8: The Main Subsystem

Describes the commands for changing the password, saving and reading environment files, turning on scroll control, and performing other session-related tasks.

Chapter 9: The Management Subsystem

Describes the commands for Port Monitoring, automatic segment-state detection, and other configuration and management features.

Chapter 10: The Boot PROM Commands

Describes the Packet Engine boot PROM commands.

Chapter 11: The NVRAM Subsystem

Describes the NVRAM configuration commands.

7 Getting Started with the User Interface

This chapter describes the user interface and how to issue commands. In addition, this chapter describes the global commands, which you can use to perform the following tasks:

- Reboot the software. (See Section 7.2 on page 124.)
- Log in. (See Section 7.4 on page 127.)
- Log out. (See Section 7.5 on page 128.)
- Access a subsystem. (See Section 7.6 on page 128.)
- Get on-line help. (See Section 7.7 on page 130.)
- Use the command history. (See Section 7.8 on page 132.)
- Define and use command aliases. (See Section 7.9 on page 134.)

7.1 GLOBAL COMMANDS

Table 7–1 lists and describes the global commands, indicates whether they are valid under root or monitor capability, and lists the section that contains information about how to use the command.

TABLE 7–1 Global commands.

Command and Description	Capability*	See…
alias|al [*<name>***]** **[***<name> <def>***]** Lists current command aliases or defines a new alias.	R or M	7.9
atalk Accesses the AppleTalk subsystem.	R or M	7.6
bridge Accesses the bridge subsystem.	R or M	7.6
bye Ends a command-line session. Equivalent to the **logout** command.	R or M	7.5
dec Accesses the DECnet subsystem.	R or M	7.6
fddi Accesses the FDDI subsystem.	R or M	7.6
findcmd|fcmd [*<command>***]** Lists the subsystem in which the specified command is located. If <*command*> is not specified, lists all commands in all subsystems.	R or M	7.7.2
help|h [*<command>***]** Displays on-line help for the specified command. If no command is specified, displays on-line help for all commands within the current subsystem.	R or M	7.7.3 7.7.4 7.7.5
history|hi Displays the command history (up to 32 commands) for the current session.	R or M	7.8
histchars|hch *<ch1>* **[***<ch2>***]** Displays or changes the history-control characters.	R or M	7.8.1 7.8.2
ip Accesses the IP subsystem.	R or M	7.6
*R= Root, M=Monitor.		

TABLE 7–1 (Continued) Global commands.

Command and Description	Capability*	See…
ipm Accesses the IP Multicast subsystem.	R or M	7.6
ipx Accesses the IPX subsystem.	R or M	7.6
logout|lo Ends a command-line session. Equivalent to the **bye** command.	R or M	7.5
main Accesses the main subsystem.	R or M	7.6
mgmt Accesses the management subsystem.	R or M	7.6
nvram Accesses the NVRAM subsystem.	R or M	7.6
ospf Accesses the OSPF subsystem.	R or M	7.6
rip Accesses the RIP subsystem.	R or M	7.6
snmp Accesses the SNMP subsystem.	R or M	7.6
subsystems|ss Displays a list of all the subsystems.	R or M	7.6
tcpstack Accesses the TCP subsystem.	R or M	7.6
tftp Accesses the TFTP subsystem.	R or M	7.6
unalias|ual *<name>* Removes a command alias.	R or M	7.9.4
*R= Root, M=Monitor.		

PowerHub commands are grouped into *subsystems*. You can access the global commands (listed in Table 7–1) from any subsystem. If you want to be able to issue additional commands (ex: **ip ping**) from any subsystem, you can create aliases for the commands as described in Section 7.9 on page 134.

7.2 REBOOTING THE POWERHUB SOFTWARE

You can boot (or reboot) the system software using any of the following methods:

- Press the reset switch (labeled `RST`), located on the front of the Packet Engine.
- Issue the **`mgmt reboot`** command. (See Section 9.3.1 on page 157.)
- Issue the **`boot`** (**`b`**) command at the `<PROM-6pe>` prompt.
- Turn the power supplies off, then back on.

The following example shows the types of boot messages you will see.

```
FORE Systems PowerHub 6000 Packet Engine
Prom version: 2.5 (s1.61) 1995.04.19 09:35
I-cache 16K OK
entering cached code
I-cache execution OK
D-cache 4K OK
SRAM 128K OK
DRAM ..............7680K OK
Shared Memory ...2048K OK
Entering Monitor
FlashInit: found 2MB Flash Memory Module
Board Type: 6PE ,  CpuType: MCPU,  Instance: 1
Ethernet address: 00-00-ef-01-e4-80

(normal start)

Hit any key now to abort boot    :      [5]:

Trying flash module boot...
Boot definition file: fm:bootdef (default)
Using disk bootdef, parsed as version 0
Loading file "fm:6pe" (AB format)
File loading complete
    ...hardware set-up messages omitted for brevity.
    ...segment set-up messages omitted for brevity.
    ...cfg file read, if present on boot source.

login:
```

7.3 ISSUING COMMANDS

Regardless of whether you are accessing the PowerHub 6000 through a TTY (RS-232) port or through a TELNET session, you issue commands at a command prompt such as the one shown in Figure 7–1:

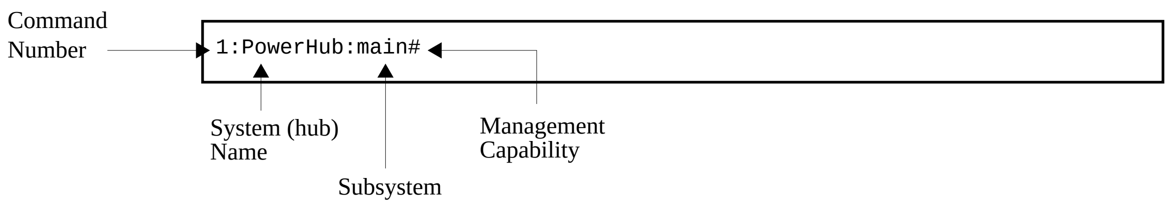


FIGURE 7–1 PowerHub command prompt.

As shown in Figure 7–1, the command prompt has four components:

Command Number — The sequential number of the command in this session (similar to a command number in the UNIX C-shell).

When you issue a carriage return, the hub attempts to execute the command you have entered at the command prompt. A message or data (if requested) is displayed, then a new command prompt is displayed.

System Name — The name assigned to this PowerHub 6000. The default name, `PowerHub`, can be changed using the **`mgmt sysname`** command. (See Section 9.3.3 on page 158.)

Subsystem — The name of the subsystem currently in use. Commands issued at the command prompt must either be global commands or commands within the current subsystem.

Management Capability

Indicates whether the session is in *monitor* capability or *root* capability:

- **>** — Indicates monitor (display only) capability. *Monitor* capability lets you display statistics and read configuration information. You cannot issue commands that change the configuration, clear statistics, or modify internal tables.
- **#** — Indicates root (configuration) capability. *Root* capability lets you issue any command.

> **NOTE**: If you start a user session and the `login:` prompt is displayed, rather than the command prompt, you must enter the appropriate password before proceeding. (See Section 7.4.)

The command prompt described in this section lets you issue system software commands. The `<PROM-6pe>` prompt also can be used to issue some commands, including NVRAM commands, but does not allow most system software commands. See Section 2.3.8 on page 28 and Section 3.1 on page 43 for information on the `<PROM-6pe>` prompt.

7.3.1 Entering and Editing Command Text

All commands are typed at the command prompt using a keyboard attached to the workstation, terminal, or PC you are using as a management station. The management station must be attached to the `TTY1` port or connected to the hub through a TELNET session.

Commands and arguments are case-sensitive and should be entered only as shown in the manual or on-line help. Each command must fit on a single line and cannot exceed 128 characters in length. The keys you use to edit and issue commands are the standard keys used on most UNIX workstations:

- To issue a command, enter the command name and arguments (if needed) after the command prompt, then press Enter or Return.
- To erase individual characters in a command, use the Backspace or Delete key, or the EraseChar character assigned in your TELNET session (usually CTRL + H).
- To cancel an entire line of input, use the reassign character (usually CTRL + U).
- To control the scrolling of output on the terminal, use CTRL + S to stop the flow and CTRL + Q to resume the flow. You can set the scroll amount using the **`main stty`** commands. (See Section 8.7 on page 143 for information.)
- You can use commands in the **tcpstack** subsystem to display or change the key sequences used in current sessions, TELNET sessions, or all sessions. For more information about these commands, see Chapter 3 in the *PowerHub Software Manual, V 2.6* (Rev C).

7.3.2 Terse Forms

For most commands and arguments, you can save keystrokes by using a terse form. A *terse form* is an abbreviation for a command or argument. You can enter either the full form or the terse form of any command or argument.

The on-line help and this book show the terse forms of commands and arguments along with the fully spelled out forms. See page iv for information about how terse forms are shown.

7.4 LOGGING IN

Unless you set the Lock Switch (or the Lock Switch jumper) on the Packet Engine to Locked, you do not need to log in to the hub to use the command-line interface. To access the command-line interface, you simply attach a modem or management terminal to the TTY1 port and boot the software. (See Section 4.3 on page 52.)

If you set the Lock Switch jumper (or the Lock Switch jumper) to Locked, the hub displays the `login:` prompt when you boot the software. You must enter "`root`" or "`monitor`," then enter a password. The password you enter depends upon the management capability you specify:

- If you specify "`monitor`," enter the monitor password.
- If you specify "`root`," enter the root password.

See Section 7.3 for information on the management capabilities. See Section 5.12 on page 103 for the procedure to change the setting of the Lock Switch jumper.

When the hub is shipped from the factory, the password for each management capability is blank. At the `password:` prompt, just press Enter. To set or change a password, use the **`main passwd`** command. For information about the **`main passwd`** command, see Section 8.4 on page 139.

7.4.1 Baud Rates

When you first boot the software, the PowerHub 6000 uses the default baud rates for its TTY ports: 9600 for TTY1 and 1200 for TTY2. The first time you access the user interface on the hub, the management terminal or modem attached to the TTY1 port must be set to 9600 baud. If you need to use a different baud rate, you must use the **`mgmt setbaud`** command to specify the baud rate you need. See Section 9.7.1 on page 182 for instructions on setting the baud rate for a TTY port.

> **NOTE**: When you set the baud rate using the **`mgmt setbaud`** command, the rate is recorded in NVRAM. However, the baud rates recorded in NVRAM are not used if the Lock Switch or the Lock Switch jumper switch is set to Unlocked. The baud rates stored in NVRAM are ignored and the default rates are used instead.

7.5 LOGGING OUT

To log out from a PowerHub session, issue one of the following commands:

- **`bye`**
- **`logout`**

These commands end the session from which the command is issued, but do not affect other user sessions on the hub. You can end all sessions on the PowerHub 6000 by powering down the hub or rebooting. (See Section 7.2.)

7.6 ACCESSING A SUBSYSTEM

The PowerHub commands are organized into subsystems. Each *subsystem* contains commands that pertain to a particular aspect of PowerHub configuration or management.

Table 7–2 lists and describes the PowerHub subsystems. The subsystems are not organized hierarchically. You can switch the focus of the command line to any subsystem by entering its name at the command prompt, then pressing Enter. Here is an example:

```
4:PowerHub:main# mgmt
5:PowerHub:mgmt#
```

Notice that the subsystem portion of the command prompt changes to indicate the current subsystem.

.

TABLE 7–2 PowerHub subsystems.

Subsystem	Use to...
atalk	Configure segments for AppleTalk routing.
bridge	Configure segments for IEEE 802.1d bridging, configure Spanning-Tree parameters, enable or disable segments, and apply filters to bridge packets.
dec	Configure segments for DECnet routing.
fddi	Display FDDI standard and proprietary MIB objects, display FDDI statistics, and adjust the T-REQ and TVX hardware timers.
ip	Configure segments for IP routing.
ipm	Configure segments for IP Multicast routing.
ipx	Configure segments for IPX routing.

TABLE 7–2 PowerHub subsystems.

Subsystem	Use to...
main	Display software versions, set passwords, access and create timed commands and environment files, change management capability, allocate DRAM, and display how long the system has been up.
mgmt	Display hardware information, configure hardware parameters, use configuration files, and monitor segment traffic.
nvram	Configure netboot parameters.
ospf	Configure the hub as an OSPF router.
rip	Configure the hub for RIP route reporting.
snmp	Configure SNMP communities and managers.
tcpstack	Configure TELNET command sequences, display and kill TELNET sessions, and display TCP and UDP tables.
tftp	Upload, download, and display files on a TFTP server.

This manual describes the subsystems used to configure and manage the hub itself, but does not describe the subsystems used to configure the hub for bridging and routing:

- For information on the subsystems for bridging, TCP, IP routing, IP Multicasting, RIP, and SNMP, see the *PowerHub Software Manual, V 2.6* (Rev C).
- For information on subsystems for AppleTalk, DECnet, and IPX routing, as well as IP security, see the *PowerHub Supplementary Protocols Manual, V 2.6* (Rev C).
- For information about the subsystem for OSPF, see the *PowerHub OSPF Addendum*.

When you begin a session, the command prompt begins in the **main** subsystem.

You also can issue a command located in a particular subsystem from within another subsystem by prefacing the command with the name of its subsystem, as shown in the following example:

```
5:PowerHub:main# mgmt showcfg
Accelerator board is present. Accelerator IOP is being used.
Installed DRAM Size: 16 MB
tty1:  not set - using 9600 baud
tty2:  not set - using 9600 baud
PE:    slot 1
PM1:   not present
PM2:   present and good
  01/01 UTP     UTP     UTP     UTP     UTP     UTP
        UTP     UTP     UTP     UTP     UTP     UTP
6:PowerHub:main#
```

In this example, the physical configuration of the PowerHub 6000 is displayed by the management (**mgmt**) subsystem's **`showcfg`** command. Note that the command prompt displayed following the output is still set to the **main** subsystem.

7.7 GETTING ON-LINE HELP

The PowerHub 6000 contains on-line help for each subsystem and command. At any time, you can display:

- A list of all the PowerHub subsystems.
- A list of all the commands in each subsystem or in a particular subsystem.
- The subsystem(s) that contains a particular command.
- Complete syntax information for a specific command.

In addition, if you issue a valid command, but with incorrect syntax, the PowerHub software automatically displays syntax information for that command.

7.7.1 Displaying a List of Subsystems

As described in Section 7.6, the PowerHub commands are organized into subsystems. At any time, you can display a list of the PowerHub subsystems by issuing the following command:

`subsystems` (or **`ss`**)

Here is the display produced by this command:

```
1:PowerHub:main# subsystems
atalk bridge dec fddi ip ipm ipx main mgmt nvram ospf rip snmp tcpstack
tftp
2:PowerHub:main#
```

To access a subsystem listed by **`subsystems`**, type the subsystem name at the command prompt, then press Enter.

7.7.2 Displaying a List of All the Commands

To display a list of all the commands in all of the subsystems, issue this command:

`findcmd` (or **`fcmd`**)

The commands are grouped by subsystem, and displayed in alphabetical order within each subsystem. The global commands are shown first.

> **NOTE**: If the information scrolls by too quickly for you to read, you can use the **`main stty`** command to control scrolling. (See Section 8.7 on page 143 for information.)

7.7.3 Displaying a List of Commands Within a Subsystem

To display a list of all the commands within a specific subsystem, do one of the following:

- Access the subsystem, then issue the following command: **help**
- Issue the following command:

<subsystem> **help|h**

where:

<subsystem>	Is the name of the subsystem for which you want to display a list of commands.

All the commands within the subsystem are listed. In the following example, this command is used to list all the commands in the **main** subsystem.

```
2:PowerHub:main# help
Main subsystem commands:

getmem | gm                          help | h
logout | lo                          passwd | pw
readenv | rdenv                      stty | sy
saveenv | svenv                      setuser | su
sysuptime | sup                      timedcmd | tc
version | ver
```

7.7.4 Listing the Subsystem That Contains a Particular Command

To list the subsystems that contain a particular command, issue the following command:

findcmd|fcmd [*<command>***]**

where:

<command>	Is the command name.

Here is an example:

```
3:PowerHub:main# findcmd showcfg
atalk: showcfg|scf
bridge: showcfg|scf
ip: showcfg|scf
ipm: showcfg|scf
ipx: showcfg|scf
mgmt: showcfg|scf
ospf: showcfg|scf
rip: showcfg|scf
snmp: showcfg|scf
tcpstack: showcfg|scf
```

In this example, all subsystems that contain the command **showcfg** are listed. The information displayed by the **showcfg** command differs depending upon the subsystem from which it is issued.

7.7.5 Displaying Syntax Information for a Command

To display complete syntax information for a specific command, issue the following command:

help|h *<command>*

where:

<command>	Specifies the command for which you want to display syntax information. You must specify either a global command or a command within the current subsystem.

Here is an example of the use of this command:

```
4:PowerHub:main# help logout
logout|lo
     closes current command-line session
```

> **NOTE**: If you specify a valid command name for *<command>*, but no help text is displayed, make sure the command is part of the current subsystem. If the command is part of a different subsystem, change to that subsystem, then reissue the **help** *<command>*.

7.8 USING THE COMMAND HISTORY

For each session, the PowerHub software maintains a history of the 32 most recently issued commands. Using the history commands, you can display the command history, reissue commands, or edit and reissue commands.

To display the 32 most recently issued commands, issue the **history** command.

To reissue or edit commands listed in the command history, use the *history control characters*. Here are the default history control characters:

!	History-prefix character.
^	Quick-substitution character.

You use the history control characters to form commands to reissue (or modify and reissue) commands from the command history. Here are the history commands you use to edit and reissue commands listed in the command history. The syntax is shown using the default history characters.

!!	Repeats the previous command.
!*<n>*	Repeats a command listed in the command history, where *<n>* indicates the number of the command as listed in the history.

!<*-i*>	Issues a previously issued command, where <*i*> is the offset back from the current command. For example, the command **!-1** gives the same results as **!!**, reissuing the previous command.
!<*substring*>	Repeats a previous command that begins with the string identified by <*substring*>.
^<*old*>**^**<*new*>	Modifies, then reissues the previous command, where <*old*> indicates the string to be replaced with <*new*>.

7.8.1 Displaying and Changing the History Control Characters

Use the **histchars** command to display the current history control characters.

To change the history control characters, issue the **histchars** command with one or both optional arguments:

histchars|hch <*hist-char*> **[**<*quick-char*>**]**

where:

<*hist-char*>	Specifies the new character you want to use in place of the current history-prefix character.
<*quick-char*>	Specifies the new character you want to use in place of the current quick character.

Here is an example of the use of this command:

```
5:PowerHub:main# histchars
history sub: !      quick sub: ^
```

7.8.2 Changing Only the Quick-Substitution Character

Although the history-prefix character is normally used only at the beginning of the command line, it is recognized and active anywhere on the command line. To turn off the special meaning of the history-prefix character, preface it with a backslash. You must do this, for example, to issue a **histchars** command that changes the quick-substitution character but not the history-prefix character. Here is an example:

```
6:PowerHub:main# hch \!^
7:PowerHub:main# !4
help logout
logout|lo
closes current command-line session
8:PowerHub:main# ^logout^sysuptime
Shows the time of last reboot or the time elapsed since last reboot
```

7.9 USING COMMAND ALIASES

The command-line interface provides an *alias* mechanism that lets you issue frequently-used commands with just a few keystrokes. Each time you need to issue the command, you can type the alias instead of the command itself. The PowerHub alias mechanism is a simplified version of the alias mechanism in the UNIX C-shell.

Aliases are local to the current command-line session. That is, they are not remembered across logins or resets of the PowerHub 6000 unless you save them to an environment file. Each user session can have up to 32 aliases. You can store aliases in an environment file by issuing the **main saveenv** *<file-name>* command (see Section 7.9.3). See Section 8.10 on page 149 for information about environment files.

In general, when you issue an alias from the command line, it must be the first item after the command prompt. However, you can enter a subsystem name before the alias; for example:

mgmt box

where:

mgmt	Is the subsystem name.
box	Is the alias.

You also can use an alias to make a frequently used command global; for example: **alias box mgmt showcfg**.

In addition to entering an alias directly from the command line, you can use an alias as part of a timed command. (See Section 8.9 on page 145 for information about timed commands.) When the timed command is activated, the command represented by the alias is issued.

7.9.1 Defining an Alias

To define an alias, issue the following command:

alias|al *<string>* *<command>*

where:

<string>	Is the string you want to type in place of the command.
<command>	Is the command (including arguments) to which you want to assign the specified alias.

For example, to define “?” as an alias for “help,” type:

```
9:PowerHub# alias ? help
Added ?:help
```

The PowerHub software acknowledges that it has added **?** to its list of aliases. To define "`hist`" as an alias for "`history`," issue the following command:

```
10:PowerHub# alias hist history
```

Note that only one level of alias substitution is performed. That is, strings within an alias definition are not checked against the alias list. For instance, in the following example, the "`?`" alias for `help` still works even though "`help`" is defined as an alias for `subsystems`.

```
11:PowerHub# alias help subsystems
12:PowerHub# help
atalk bridge dec fddi ip ipm ipx main mgmt nvram rip snmp tcpstack tftp

13:PowerHub# ?
Global commands:
   history|hi                  show command history
...help listing continues
```

7.9.2 Displaying an Alias

To display the definition of an alias, issue the following command:

alias|al [*<string>***]**

where:

<string>	Is the alias for which you want to display the definition. If you do not specify an alias, all aliases defined for the current session are displayed.

Here is an example of the display produced by this command:

```
14:PowerHub:main#  alias
?        help
help   subsystems
```

7.9.3 Saving and Loading an Alias

Aliases apply to the current command-line session only. For example, if you define aliases within a TTY1 session, then open a TELNET session, the aliases are not available to the new session. Moreover, if you end the current session without saving the aliases defined during that session, they are lost.

There are several ways to save aliases. The easiest way is to save them to an environment file using the following command:

main saveenv *<file-name>*

You also can manually add aliases to a file, then type **main readenv** *<file-name>* to read (load) them in each time you log in. Note that environment files contain other session parameters in addition to aliases. For information about environment files, see Section 3.2 on page 44 and Section 8.10 on page 149.

Alternatively, you can place the aliases in an external file on your management station, then load them into the command-line session each time you log in. For example, if your management terminal is a PC running Kermit, you can put them in an "`aliases`" text file on the PC, then load them into the PowerHub 6000 by escaping into the Kermit prompt and typing "`transmit aliases`". The **`transmit aliases`** command transmits the text file as if you were typing it. On UNIX systems running the `tip` program, you can use the "`~>`" escape to send a local text file to the hub.

7.9.4 Deleting an Alias

To delete an alias, issue the following command:

`unalias|ual [`*`<string>`***`]`**

where:

`<string>` Is the alias you want to delete.

Here is an example of the use of this command:

```
15:PowerHub:main# unalias ?
```

8 The Main Subsystem

This chapter describes the **main** subsystem commands, which you can use to perform the following tasks:

- Display the installed PROM and system software versions. (See Section 8.3 on page 139.)
- Change the root or monitor password. (See Section 8.4 on page 139.)
- Change management capability (from root to monitor, or from monitor to root). (See Section 8.5 on page 141.)
- Allocate memory for an optional protocol or the Bridge MIB. (See Section 8.6 on page 141.)
- Set screen display (stty) parameters. (See Section 8.7 on page 143.)
- Display the time elapsed since the most recent reboot. (See Section 8.8 on page 145.)
- Define and activate timed commands. (See Section 8.9 on page 145.)
- Save or read (load) an environment file. (See Section 8.10 on page 149.)

8.1 ACCESSING THE MAIN SUBSYSTEM

To access the **main** subsystem, issue the following command at any command prompt:

```
main
```

8.2 MAIN SUBSYSTEM COMMANDS

Table 8–1 lists and describes the **main** subsystem commands. For each command, the table indicates whether it is valid under root or monitor capability (explained in Section 7.3 on page 125), and lists the section in this chapter that contains information about the command.

TABLE 8–1 Main subsystem commands.

Command and Description	Capability*	See…
getmem|gm ipx|dec|atalk|brmib **getmem|gm ospf [***<Kbytes>***]** **getmem|gm ipm [4k|8k]** Allocates memory for the specified routing subsystem or MIB. If you need to use this command, issue it as soon as you boot the software.	R	8.6
passwd|pw [root|monitor] Changes the password for root or monitor capability.	R	8.4.1
readenv|rdenv *<file-name>* Reads the operating environment from the specified file.	R or M	8.10.2
saveenv|svenv *<file-name>* Saves the current operating environment to the specified file.	R or M	8.10.1
setuser|su [root|monitor] Sets the management capability of the current user session.	R or M	8.5
stty|sy [rows *<numrows>***] | [more|mo enl|dis]** Sets the scroll amount and enables the “more” feature.	R or M	8.7
sysuptime|sup Shows how much time has elapsed since the system software was booted.	R or M	8.8
timedcmd|tc **timedcmd|tc add** *<id> <interval(secs)> <cmd-and-args>* **timedcmd|tc on|off|del** *<id>* Used without arguments, displays all timed commands and their status. Used with **add**, adds the specified timed command. Used with **on|off|del**, enables, disables, or deletes the specified timed command.	R or M	8.9
version|ver [*<slot-number>***|all]** Displays the installed versions of software.	R or M	8.3
***R= Root, M=Monitor.		

8.3 DISPLAYING THE INSTALLED SOFTWARE VERSIONS

To display the software versions running on the PowerHub system, issue the following command:

main version [*<slot-number>***|all]**

where:

*<slot-number>***|all**	Specifies either a specific NIM slot or all NIM slots in the chassis. If you do not specify a slot number, the command displays the software versions (system software and boot PROM) installed on the Packet Engine.

Here is an example of the information shown by this command. In this example, the **version** command is issued without an argument.

```
1:PowerHub:main# version
############## Slot 1 ##############

       Card Type: Packet Engine
        Serial #: 538027795
           Model: 6101-00
        Revision: B
           Issue: 1
PowerHub Version: 6-2.6.3.0 (s1.282) 1996.03.25 16:56
    PROM Version: 6pep-2.5 (s1.61) 1995.04.19 09:35
```

This example shows that the hub contains the following software:

- System software version 6-2.6.3.0.
- Packet Engine boot PROM version 2.5.

The numbers in parentheses following the software version names are used by FORE Systems TAC and might differ from the number shown in this example.

Notice that the date and time when the final versions of the software were officially released by the factory are listed to the right of the software versions. (The release dates and times shown in this example might differ from those actually displayed.)

If you need to upgrade some or all of the software installed on your system, see the *PowerHub 6000 Release Notes* or contact FORE Systems TAC.

8.4 SECURING ACCESS TO THE POWERHUB 6000

As shipped from the factory, the PowerHub 6000 automatically begins a command-line session on the modem or management terminal attached to the `TTY1` port. By changing the setting of the Lock Switch jumper, you can reconfigure the software to require a login ID (“`root`” or “`monitor`”) and a password at the beginning of each session. Thus configured, the hub does not accept any commands until a valid password is provided for the requested login.

To ensure secured access to the PowerHub 6000:

- Install the hub in a secure room and control access to the room.
- Make sure the Lock Switch (or Lock Switch jumper) on the Packet Engine is set to the Locked position. (See Section 5.12 on page 103.)
- Change the password for root and monitor access. When the hub is shipped from the factory, the password for each management capability is blank. Section 8.4.1 describes how to change a password.

8.4.1 Changing a Login Password

You can change the password for the "`root`" or "`monitor`" login ID using the **`passwd`** command. Here is the syntax for this command:

`passwd|pw [root|monitor]`

where:

`root|monitor`	Indicates the management capability for which you are changing the password. The default is **`root`**.

The following example shows how a password for root management capability is changed.

```
31:PowerHub: passwd root
Old password: root
New password: Lab1mojo
Re-enter new password: Lab1mojo
Password changed
```

After you enter the new password, you are prompted to enter the old password, and re-enter the new password.

For security, the input shown with italicized characters in this example does not appear when actually typed on the screen.

> **NOTE**: The `Old password:` prompt is not displayed if the Lock Switch or the Lock Switch jumper is set to Unlocked. Instead, the `New password:` prompt is displayed. Thus, if you forget the password, you can unlock the Lock Switch, log in, enter a new password, then reset the Lock Switch or the Lock Switch jumper to the Locked position. (See Section 5.12 on page 103.)

8.5 CHANGING ACCESS LEVEL (MANAGEMENT CAPABILITY)

To change your management capability during a session, issue the following command at the command prompt:

setuser|su [root|monitor]

where:

root|monitor	Specifies whether you are changing the user session to root capability or monitor capability. The default is **root**.

Here is an example of how to change the management capability from root to monitor. Note that the management-capability portion of the command prompt changes from # to >.

```
32:PowerHub:main# setuser monitor
password: <Password entered here is not shown on screen.>

33:PowerHub:main>
```

To return to the previous management capability, issue the **logout** (**lo**) command. (You cannot "nest" **setuser** commands.) Here is an example:

```
33:PowerHub:main> logout
34:PowerHub:main#
```

8.6 ALLOCATING MEMORY FOR OPTIONAL PROTOCOLS

The default configuration for the PowerHub 6000 boots with a portion of the Packet Engine main memory (DRAM) allocated to general bridging and routing functions. This memory is sufficient for standard bridging and IP routing. However, if you plan to use any of the following additional protocols or features, you must allocate memory specifically for their use:

- AppleTalk
- DECnet
- IPX
- Bridge MIB IP Multicasting
- OSPF

NOTE: We recommend that you allocate memory for additional protocols or features before you begin bridging or routing. Otherwise, the necessary memory might not be available when you need it.

Depending upon the amount of memory in your system, you can allocate memory for some or all of the features listed above. (See Section 2.3.7 on page 28.)

To allocate memory for AppleTalk, DECnet, IPX, or the Bridge MIB, issue the following command:

`getmem|gm ipx|dec|atalk|brmib`

where:

`ipx|dec|atalk|brmib`

Specifies the protocol or MIB for which you are allocating memory:

`ipx`	IPX routing protocol.
`dec`	DECnet routing protocol.
`atalk`	AppleTalk routing protocol.
`brmib`	Bridge MIB.

> **NOTE**: The memory remains allocated until you reboot. To cause memory to automatically be allocated when the PowerHub 7000 boots, allocate memory, then save the current configuration.
>
> You can allocate memory for a specific protocol (such as AppleTalk) only one time during a power cycle. For example, after the **`main getmem atalk`** command is issued once within a power cycle (either from within the PowerHub configuration file or by you), the command does not affect the amount of DRAM allocated to AppleTalk.
>
> To change the amount of DRAM allocated to a protocol or the Bridge MIB, issue the **`main getmem`** command again (specifying the amount of DRAM you want to allocate, if applicable), save the configuration file, then reboot the hub. When you reboot, the hub reads the new DRAM amount from the configuration file.
>
> See Section 9.8.2 on page 188 for information about saving the configuration file.

To allocate memory for IP Multicasting, issue the following command:

`getmem|gm ipm [4k|8k]`

where:

`4k|8k`	Specifies the size of the IP Multicasting route table. If you do not specify a size, the default allocation (2k) is used.

To allocate memory for OSPF routing, issue the following command:

getmem|gm ospf [*<Kbytes>***]**

where:

<Kbytes>	Specifies how many KB of the Packet Engine's main memory you want to allocate to OSPF. You can specify an amount from **10** through **500** KB, in 1-KB increments. The default is **500** KB. FORE Systems recommends that you try the default allocation first. Let the PowerHub system act as an OSPF router in your OSPF network for a few hours, then use the **ospf stats** command to check the memory usage. If the `Free` field shows more than 25% of the memory is unused, you might want to change the memory allocation to a smaller amount. (See Chapter 3 in the *PowerHub OSPF Addendum*.) In general, you want to allocate enough memory so that about 25% is free. If the **ospf stats** command indicates memory allocation failures (in the `Total Authorization Failures` field), you need to allocate more memory to OSPF.

8.7 SETTING THE SCROLL (STTY) PARAMETERS

Use the **stty** commands to control the scrolling of data displayed on the management terminal. These commands let you:

- Enable or disable the "more" feature, which controls scrolling of information on the screen.
- Specify the maximum number of rows displayed on the management terminal at one time when the "more" feature is enabled.

8.7.1 Enabling or Disabling the "More" Feature

To enable or disable the "more" feature, issue the following command:

stty|sy more|mo [enl|dis]

where:

enl|dis	Enables or disables the "more" feature. When this feature is disabled, the software does not control scrolling.

If you do not specify **enl** or **dis**, the current scroll control settings are displayed, as shown in the following example.

```
PowerHub:main# mgmt more
Number of rows on screen set to : 24
'more' is enabled
```

When the "more" feature is enabled, output that exceeds the number of rows you specified is paused until you press enter or the space bar to continue, or type "q" to end the output. This is shown in the following example:

```
PowerHub:main# mgmt showfile root.env
#
# stty
#
main stty rows 24
main stty more dis

#
# aliases
#
main alias aarp   atalk at *.1
main alias br     bridge s all pi,po
main alias bru    bridge s all pu,cu
main alias gilligan  ip ping 181.17.45.17
main alias skipper   ip ping 191.1.45.3
main alias marianne  ip ping 131.24.45.2
main alias ginger    ip ping 131.24.45.2
-- more [<space>,<ret>,q] --
```

As shown in this example, you can respond to the `more` prompt with any one of the following:

Press the Space Bar	Scrolls the display one screen-full forward.
Press Return	Scrolls the display one line forward.
Press **q**	Ends the display.

8.7.2 Setting the Scroll Amount

To specify the maximum number of rows that can be displayed on the terminal, issue the following command. Note that regardless of the number of rows you specify, the software controls scrolling only if you enable the "more" feature, as described in Section 8.7.1.

stty|sy [rows *<numrows>***]**

where:

rows [*<numrows>***]**	Specifies the number of rows of data that are displayed at one time. If a file or table contains more rows than the number you specify, the PowerHub 6000 pauses until you are ready to display the next group of rows. If you issue the **stty** command without the **rows** argument, the current scroll control settings are displayed.

Here is an example of how to use this command.

```
34:PowerHub:main# stty rows 20
Number of rows on screen set to 20
```

8.8 DISPLAYING TIME ELAPSED SINCE LAST REBOOT

Use the **`sysuptime`** command to display the amount of time that has elapsed since the last reboot of the system software. Here is an example of the display produced by this command:

```
38:PowerHub:main# sysuptime
Elapsed time since last reboot: 217 hours, 33 minutes, 24 seconds
```

8.9 USING TIMED COMMANDS

When you use a command-line session to monitor network behavior, you might want to execute certain commands regularly and repeatedly. For example, you might want to display a bridge or route cache at regular intervals to observe frequently requested bridge or route destinations for certain segments.

You can define and activate a timed command to automatically issue any command string at a regular interval. A *timed command* is similar to an alias, but is automatically issued by the PowerHub 6000 at an interval you specify. You can define PowerHub 6000 commands and even aliases as timed commands. Each user session can have up to eight timed commands.

Table 8–2 lists the commands used to define, display, activate, and delete timed commands.

TABLE 8–2 Timed commands.

Use command...	**To...**
`timedcmd|tc`	Display all timed commands.
`timedcmd|tc add` *`<id> <interval(secs)> <cmd-and-args>`*	Add a timed command.
`timedcmd|tc on` *`<id>`*	Start the timer for a timed command.
`timedcmd|tc off` *`<id>`*	Stop the timer for a timed command.
`timedcmd|tc del` *`<id>`*	Delete a timed command.

As with command aliases (see Section 7.9 on page 134), timed commands are local to the current command-line session. That is, they are not remembered across logins, at the close of command-line sessions, or resets of the PowerHub 6000 unless you save them to an environment file.

To save timed commands to an environment file, issue the **`main saveenv`** *`<file-name>`* command. See Section 8.10 for information about environment files.

8.9.1 Defining a Timed Command

Use the **`timedcmd add`** command to define a timed command. Here is the syntax for this command:

`timedcmd|tc add` *`<id> <interval(secs)> <cmd-and-args>`*

where:

`<id>`	Specifies the name of the timed command. When you activate the timed command, you use this name. You can specify an alphanumeric string up to 15 characters in length.
`<interval(secs)>`	Specifies, in seconds, the interval at which the timed command is reissued. You can specify a minimum of `1` second.
`<cmd-and-args>`	Specifies the command string that is issued each time the interval specified by *`<secs>`* expires. You can specify a PowerHub command, including its arguments, or an alias.

> **NOTE**: You must include the subsystem name in each timed command. For example, if you create a timed command that issues the **`interface-table`** command from within the **atalk** (AppleTalk) subsystem, specify the command as **`atalk interface-table`** (or use the terse form: **`atalk it`**). Otherwise, you might get unexpected results.

Here is an example of how to define a timed command. In this example, a timed command named "**`bcache`**" is defined to automatically display the bridge cache every 10 seconds. A command such as this is useful for quickly observing bridge activity.

```
35:PowerHub:main# timedcmd add bcache 10 bridge display-cache 1-6
Added bcache: 10 secs, bridge display-cache 1-6  (timer not running)
```

For information about the **bridge** subsystem and bridge cache, see Chapter 2 in the *PowerHub Software Manual, V 2.6* (Rev C).

8.9.2 Starting a Timed Command

To use a timed command, you must start the timer for the command. When you start the timer, the command is issued when the specified timer value expires. The PowerHub 6000 continues to issue the timed command each time the interval expires until you issue the **`timedcmd off`** command (discussed in Section 8.9.3), or until you log out. Note that if you issue the **`main saveenv`** *`<file-name>`* command while the timed command is running, the **`timedcmd on`** *`<id>`* command is added to the environment file. Consequently, the timed command is started again the next time you read the environment file specified by *`<file-name>`*.

Use the **`timedcmd on`** command to start the timer for a timed command. Here is the syntax for this command:

`timedcmd|tc on` *<id>*

where:

<id>	Specifies the name of the timed command for which you are starting the timer. Make sure you specify the name of the timed command itself, rather than the command string associated with the timed command.

Here is an example of the use of this command. The output is produced by the timed command **`bcache`**, defined in the example in Section 8.9.1.

```
36:PowerHub:main# timedcmd on bcache
bcache: started at 10 seconds interval
Bridging cache:
Port 01:  empty
Port 02:  empty
Port 03:  Dest:  08-00-20-08-70-54, Source: 08-00-20-0f-dd-99
          Dest:  06-cc-df-08-bb-aa, Source: 08-00-20-0f-dd-99
          Dest:  08-00-20-08-70-54, Source: 08-aa-20-bb-dd-99
Port 04:  empty
Port 05:  empty
Port 06:  empty


Bridging cache:
Port 01:  empty
Port 02:  empty
Port 03:  Dest:  08-00-20-08-70-54, Source: 08-00-20-0f-dd-99
          Dest:  06-cc-df-08-bb-aa, Source: 08-00-20-0f-dd-99
          Dest:  08-00-20-08-70-54, Source: 08-aa-20-bb-dd-99
          Dest:  08-00-20-08-70-54, Source: 08-aa-20-bb-dd-99
Port 04:  empty
Port 05:  empty
Port 06:  Dest:  aa-bb-cc-dd-ee-ff, Source: 01-02-04-04-04-06

timed command issued repeatedly until stopped by user...
```

Note that the timed command (**`bcache`**) in this example is issued repeatedly by the system. The different displays produced each time the command is issued reflect the changing contents of the bridge cache. For information about the **bridge** subsystem and bridge cache, see Chapter 2 in the *PowerHub Software Manual, V 2.6* (Rev C).

8.9.3 Stopping a Timed Command

Use the **`timedcmd off`** command to stop a command timer. When you stop a command timer, the PowerHub software stops issuing the timed command. Here is the syntax for the **`timedcmd off`** command.

`timedcmd|tc off` *`<id>`*

where:

`<id>`	Specifies the name of the timed command. Make sure you specify the name of the timed command itself, rather than the command string associated with the timed command.

Alternatively, you can stop a timed command by ending the command-line session from which the timed command was started.

You can restart the timed command by issuing the following command: **`timedcmd on`** *`<id>`*.

Alternatively, if you save an environment file while the timed command is running, or manually add the timed command (and the **`timedcmd on`** *`<id>`* command) to the environment file, the timed command starts again when you read (load) the environment file. See Section 8.10 on page 149 for information about environment files.

8.9.4 Deleting a Timed Command

Use the **`timedcmd del`** command to delete a timed command. Here is the syntax for this command:

`timedcmd|tc del` *`<id>`*

where:

`<id>`	Specifies the name of the timed command you want to delete.

Here is an example of the use of this command:

```
37:PowerHub:main# timedcmd del bcache
bcache: deleted
```

8.10 USING ENVIRONMENT FILES

At any time during a command-line session, you can save or read (load) an *environment file*. An environment file is an ASCII file that contains PowerHub 6000 commands defining the following parameters:

- Scroll (stty) parameters (maximum number of rows and "more" enable or disable; defined using the **`stty`** command).
- Command aliases (created using the **`alias`** command).
- Timed commands (created using the **`timedcmd`** command).

Normally, when you end a session by logging out or rebooting, any changes made to these environment settings are lost. However, if you save an environment file before logging out or rebooting, environment settings are placed into an environment file. If you read (load) that file during a command-line session, the commands in the file recreate the scroll parameters, command aliases, and timed commands that were active when you saved the file.

Here is an example of an environment file:

```
#
# stty
main stty rows 24
main stty more enl

#
# aliases
#
main alias aarp   atalk at *.1
main alias br     bridge s all pi,po
main alias bru    bridge s all pu,cu
main alias gilligan  ip ping 181.17.45.17
main alias skipper   ip ping 191.1.45.3
main alias marianne  ip ping 131.24.45.2
main alias ginger    ip ping 131.24.45.3

#
# timed commands
#
main timedcmd add bcache 10 bridge display-cache 1-6
```

Notice that the file contains three sections: `stty`, `aliases`, and `timed commands`, shown in bold type in the example. The sections are labeled by comment lines, which begin with `#`. The `stty` section contains commands to set the "more" feature and specify the maximum number of rows to be displayed when the "more" feature is enabled. The `alias` section contains **`alias`** commands that define various aliases. In this example, aliases are created to display tables and packet statistics, and to ping IP addresses. The `timed commands` section contains a command that defines the timed command **`bcache`**, described in the examples in Section 8.9.

You can save or load multiple environment files during a command-line session; the effects are cumulative. Note, however, that earlier settings can be overwritten by later settings. For example, if you read a file that contains the **`stty more enl`** command (to enable the "more" feature), then read another file that contains the **`stty more dis`** command, the net effect will be that "more" is disabled. If you experience unexpected results when using multiple environment files, examine the files to ensure that you are not inadvertently overwriting some parameters.

You can cause an environment file to be automatically loaded at the beginning of a command-line session by saving the file as one of the following:

`root.env`	If the PowerHub 6000 is booted from the Flash Memory Module, and either the Lock Switch is Disabled or you log in under root capability, this file is loaded, if present.
`monitor.env`	If the PowerHub 6000 is booted from the Flash Memory Module, the Lock Switch is Enabled, and you log in under monitor capability, this file is loaded, if present.

> **NOTES**: If you choose to manually edit an environment file, do not place any commands associated with using the Flash Memory Module into the file. This includes commands such as **`readenv`** and **`readcfg`**, which read files.
>
> Also, if the environment file contains the **`timedcmd on`** command and the timed command it starts has been defined in the environment file or earlier in the user session, the timed command is started when the environment file is read.

The following sections describe how to save the environment settings and how to activate the settings during a user session.

8.10.1 Saving an Environment File

To save an environment file, issue the following command:

`saveenv|svenv` *`<file-name>`*

where:

`<file-name>`	Specifies the name of the environment file. You can specify up to eight alphanumeric characters plus a three-character extension. Use a period to separate the name from the extension. We recommend that you always use the extension `env` (for example: `Lab1.env`) to distinguish environment files from other files.

> **NOTE**: If you are in monitor mode, you cannot save to the file name `root.env`. You are in monitor mode if you logged on to the user session as "`monitor`."

Here is an example of how to save a default environment file. In this example, the command-line session is in monitor capability. Correspondingly, the current environment settings are saved into a file called `monitor.env`.

```
32:PowerHub:main> saveenv monitor.env
monitor.env: Environment saved
```

In the following example, the scroll parameters are set, then some aliases and a timed command are defined. These environment settings are then saved into an environment file named `Lab1.env`. At any time during a command-line session, this file can be read (loaded) to activate the environment settings it contains.

Note that ***all*** current environment settings (aliases, timed commands, and scroll parameters) are saved in the environment file. These commands create the environment file shown in the following example. As you can see, environment files can save you a lot of typing.

```
33:PowerHub:main# stty rows 24
34:PowerHub:main# stty more dis
35:PowerHub:main# alias aarp   atalk at *.1
36:PowerHub:main# alias br     bridge s all pi,po
37:PowerHub:main# alias bru    bridge s all pu,cu
38:PowerHub:main# alias gilligan  ip ping 181.17.45.17
39:PowerHub:main# alias skipper   ip ping 191.1.45.3
40:PowerHub:main# alias marianne  ip ping 131.24.45.2
41:PowerHub:main# alias ginger    ip ping 131.24.45.2
42:PowerHub:main# timedcmd add bcache 10 bridge display-cache 1-6
43:PowerHub:main# saveenv monitor.env
Lab1.env: Environment saved
```

8.10.2 Reading (Loading) an Environment File

You can load an environment file and thereby activate the environment settings saved in that file using the following command:

readenv|rdenv *<file-name>*

where:

<file-name>	Specifies the name of the file that contains the environment settings. If the file is found, the commands in the file are executed; otherwise, an error message is displayed.

8.10.3 Editing an Environment File

As an alternative to setting environment parameters, then using **saveenv** to create or edit an environment file, you can edit the file directly using a text or ASCII editor. See Appendix C for a detailed description of the ways you can edit an ASCII file.

> **NOTE**: Do not place any commands associated with using the Flash Memory Module in the file. This includes commands such as **readenv** and **readcfg**, which read files.

9 The Management Subsystem

This chapter describes the commands in the management (**mgmt**) subsystem. The management commands control the configuration of the PowerHub 6000 and its connections to your network segments.

You can use the management commands to perform the following tasks:

- Reboot the software. (See Section 9.3.1 on page 157.)
- Display the chassis configuration, boot source, and MAC-layer hardware address. (See Section 9.3 on page 157.)
- Display or set the system name, location, time, and date. (See Section 9.3 on page 157.)
- Display ID and power information or the current temperature for the Packet Engine or a NIM. (See Section 9.4 on page 163.)
- Configure the traffic LEDs on the Ethernet segments. (See Section 9.4.3 on page 164.)
- Assign a name to a segment; activate an EMA (Ethernet Media Adapter) or AUI Media Cable, or re-activate a UTP connector; enable or disable automatic segment-state detection for a segment. (See Section 9.5 on page 165.)
- Display state-change statistics for a segment. (See Section 9.5.5 on page 172.)
- Manage files in the Flash Memory Module. (See Section 9.6 on page 176.)
- Set the baud rate for the TTY ports, or open or close a session on TTY2. (See Section 9.7 on page 181.)
- Create, edit, and load configuration files. (See Section 9.8 on page 183.)
- Monitor segment traffic using the Port Monitoring feature. (See Section 9.9 on page 193.)

9.1 ACCESSING THE MANAGEMENT SUBSYSTEM

To access the **mgmt** subsystem, issue the following command at any command prompt:

```
mgmt
```

9.2 MANAGEMENT SUBSYSTEM COMMANDS

Table 9–1 lists and describes the **mgmt** subsystem commands and their syntax. For each command, the management capability (explained in Section 7.3 on page 125) is listed, as well as the section that contains additional information about the command.

TABLE 9–1 Management subsystem commands.

<table>
<tr><th>Command and Description</th><th>Capability*</th><th>See…</th></tr>
<tr><td>activate-ema|ae <seg-list>
Activates an EMA or AUI Media Cable in systems containing the UMM.</td><td>R</td><td>9.5.3</td></tr>
<tr><td>activate-utp|au <seg-list>
Activates a UTP connector bypassed by an EMA or AUI Media Cable. Applies only to systems containing the UMM.</td><td>R</td><td>9.5.2</td></tr>
<tr><td>autoportstate|aps
[<seg-list>|all enl|dis [<threshold>]]
Enables or disables automatic segment-state detection.</td><td>R or M #</td><td>9.5.4</td></tr>
<tr><td>bootinfo|bi
Indicates the boot source (Flash Memory Module or TFTP server) used by the hub.</td><td>R or M</td><td>9.3.2</td></tr>
<tr><td>checksum|sum <filespec> [<filespec>...]
Computes and displays a checksum for the specified file(s).</td><td>R or M</td><td>9.6.7</td></tr>
<tr><td>copy|cp <src-file> <dest-file>
Makes a copy of the specified file on the Flash Memory Module.</td><td>R</td><td>9.6.4</td></tr>
<tr><td>date|da [<yymmddhhmm>[.<ss>]]
Displays or sets the system time and date.</td><td>R or M #</td><td>9.3.4</td></tr>
<tr><td>dir [<filespec>]
Lists directory and volume information for the specified file(s) on the Flash Memory Module.</td><td>R or M</td><td>9.6.2</td></tr>
<tr><td colspan="3">*R= Root, M=Monitor
R=display and manipulate, M=manipulate only.</td></tr>
</table>

TABLE 9–1 (Continued) Management subsystem commands.

<table>
<tr><th>Command and Description</th><th>Capability*</th><th>See…</th></tr>
<tr><td>endcfg|ecfg
Marks the end of a configuration file. This command applies only to configuration files and must be the last command in the file.</td><td>NA</td><td>9.8.1</td></tr>
<tr><td>ethaddr|ea
Displays the MAC-layer hardware address of the PowerHub 6000.</td><td>R or M</td><td>9.3.5</td></tr>
<tr><td>format|fmt <device-name>
Re-formats the Flash Memory Module.</td><td>R</td><td>9.6.8</td></tr>
<tr><td>fremove|frm [-f] [-i] <filespec> [<filespec>...]
Removes a file from the Flash Memory Module.</td><td>R</td><td>9.6.6</td></tr>
<tr><td>idprom|idp <slot-num>
Displays identification information and power requirements for the Packet Engine or NIM in the specified chassis slot.</td><td>R or M</td><td>9.4.1</td></tr>
<tr><td>led-config|lc [<slot-num> [ca|xr]]
Displays or changes the configuration of the traffic LEDs on Ethernet segments.</td><td>R or M #</td><td>9.4.3</td></tr>
<tr><td>listdir|ls [<filespec>]
Displays directory data for the specified file(s) on the Flash Memory Module.</td><td>R or M</td><td>9.6.2</td></tr>
<tr><td>operating-mode|om
[<seg-list>|all [fdx|normal]]
Changes the operating mode (half-duplex or full-duplex) on the specified Ethernet segments.</td><td>R or M #</td><td>9.5.6</td></tr>
<tr><td>port-aps-down-count|pad
<seg-list> show|s|clear|c [-t]
Lists or clears statistics for the number of times the automatic segment-state detection feature has detected that the specified segment is down and accordingly taken the segment out of service.</td><td>R or M #</td><td>9.5.5.1</td></tr>
<tr><td>port-aps-up-count|pau
<seg-list> show|s|clear|c [-t]
Lists or clears statistics for the number of times the automatic segment-state detection feature has detected that the specified segment is up and accordingly brought the segment into service.</td><td>R or M #</td><td>9.5.5.2</td></tr>
<tr><td>port-mgmt-down-count|pmd
<seg-list> show|s|clear|c [-t]
Lists or clears statistics for the number of times the specified segment has explicitly been put out of service by a user command.</td><td>R or M #</td><td>9.5.5.3</td></tr>
<tr><td colspan="3">*R= Root, M=Monitor
R=display and manipulate, M=manipulate only.</td></tr>
</table>

TABLE 9–1 (Continued) Management subsystem commands.

Command and Description	Capability*	See…
port-mgmt-up-count|pmu *<seg-list>* **show|s|clear|c [-t]** Lists or clears statistics for the number of times the specified segment has explicitly been put into service by a user command.	R or M #	9.5.5.4
port-monitor|pm view *<options>***|all** *<seg-list>* **on** *<seg-list>* **port-monitor|pm viewpair** *<seg1> <seg2>* **on** *<seg-list>* **[r]** **port-monitor|pm close** **port-monitor|pm** The Port Monitoring commands. These commands monitor traffic on one or more PowerHub segments and display the results on a protocol analyzer attached to a PowerHub segment.	R	9.9
readcfg|rdcfg *<file-or-device-name>* Reads the specified configuration file or loads a configuration file from the specified device.	R	9.8.3
reboot Reboots the PowerHub 6000 software.	R	9.3.1
remove|rm [-f] [-i] *<filespec>* **[***<filespec>...***]** Removes a file from the Flash Memory Module.	R	9.6.6
rename|ren *<oldfilename> <newfilename>* Renames the specified file on the Flash Memory Module.	R	9.6.5
savecfg|svcfg [*<file-or-device-name>***]** Saves the current PowerHub configuration to the specified file or device.	R	9.8.2
setbaud|sb tty1|tty2 *<baud-rate>* Sets the baud rate for the specified TTY port.	R	9.7.1
set-portname|spn *<seg> <name>* Renames a PowerHub segment.	R	9.5.1
showcfg|scf Displays configuration information for the PowerHub chassis.	R or M	9.3.6
showfile|shf *<file-name>* Displays the contents of a file stored on the Flash Memory Module.	R or M	9.6.3
syslocn|sl [*<location-string>***]** Defines or displays a text string describing the location of the PowerHub 6000.	R or M #	9.3.3

***R= Root, M=Monitor

R=display and manipulate, M=manipulate only.

TABLE 9–1 (Continued) Management subsystem commands.

Command and Description	Capability*	See…
sysname|sn [*<name>*] Defines or displays the PowerHub name. This name appears in the runtime command prompt.	R or M #	9.3.3
temperature|temp *<slot-num>***|all** Displays the temperature of the inside of the chassis.	R or M	9.4.2
tty2open|t2o Opens a management session on the diagnostic TTY port, TTY2.	R	9.7.2
tty2close|t2c Closes a management session on the diagnostic TTY port, TTY2.	R	9.7.3
***R= Root, M=Monitor # R=display and manipulate, M=manipulate only.		

9.3 SYSTEM MANAGEMENT COMMANDS

The system management commands configure or display information about the entire hub. You can use these commands to perform the following tasks:

- Reboot the software (**reboot** command).
- List the boot source used by the hub (**bootinfo** command).
- Set or display the system (PowerHub) name and location (**sysname** and **syslocn** commands).
- Set or display the system time and date (**date** command).
- Display the Packet Engine (and by extension the hub) MAC-layer hardware address (**ethaddr** command).
- Display the chassis configuration (**showcfg** command).

9.3.1 Rebooting the PowerHub Software

You can reboot the PowerHub 6000 software from a command-line session using the **reboot** command. This command performs a cold restart of the hub. During a cold restart, the Packet Engine conducts a power-on self-test to check its various hardware components. Following successful completion of the power-on self-tests, the hub boots the software.

For an example of the boot messages displayed by the PowerHub 6000 and additional methods for rebooting, see Section 7.2 on page 124.

9.3.2 Displaying the Boot Source Used by the Hub

You can boot the PowerHub 6000 software using files in the Flash Memory Module or on a TFTP file server. After the software is booted, it retains the following information about how it was booted:

- The system time at startup.
- The boot order specified in NVRAM (using the **nvram set bo** command; see Section 4.3.4 on page 60 or Section 11.4 on page 214).
- The boot device used by the hub to boot. The value can be `m` (Flash Memory Module) or `n` (network). This value shows the boot source actually used, rather than the boot order specified in NVRAM.
- The parser version used in the boot definition (`bootdef`) file. The parser version is always 1.
- The name of the system software "image" file that was loaded. (See Section 3.2 on page 44.)

Here is an example of the display produced by the **bootinfo** command.

```
46:PowerHub:mgmt# bootinfo
Wed Jul 13 14:41:10 1994 start
Wed Jul 13 14:41:10 1994 nvram boot order: mn
bootdevice: m
Wed Jul 13 14:41:11 1994 disk bootdef parser version used: 1
Wed Jul 13 14:41:11 1994 disk image file name: 6pe
```

9.3.3 Setting and Displaying the System Name and Location

The system name is shown in the command prompt. The default system name for each hub is `PowerHub`, as shown in the example below. You can change the system name using the following command:

sysname|sn [*<name>*]

where:

<name>	Specifies the PowerHub name. You can specify any alphanumeric string up to 24 characters in length. The name cannot contain blanks. If you do not specify a name, the current name is displayed.

The following example shows how to display, then change the system name of the hub.

```
68:PowerHub:mgmt# sysname
Current system name is: PowerHub

69:PowerHub:mgmt# sysname Buck

70:Buck:mgmt# sysname
Current system name is: Buck
```

In addition to setting the system name, you also can set the system location using the following command:

syslocn|sl [*<location-string>***]**

where:

<location-string>	Specifies the location of the PowerHub 6000. You can specify any alphanumeric string up to 24 characters in length. The location name cannot contain blanks. If you do not specify a location, the current location name is displayed.

In the following example, the system location is defined as “MonkeyHouse.” (Note that the system name is now “Buck,” as changed by the **sysname** command in the example above.)

```
71:Buck:mgmt# syslocn MonkeyHouse
72:Buck:mgmt# syslocn
Current system location is: MonkeyHouse
```

Note that the system name and location are available to SNMP-based management stations. See Chapter 8 in the *PowerHub Software Manual, V 2.6* (Rev C) for information about configuring the hub for SNMP. For a description of the name, location, and other system-related MIB objects, see Appendix A in the same manual.

9.3.4 Setting and Displaying the System Time and Date

Use the date command to display or set the system time and date. Here is the syntax for this command:

date|da [*<yymmddhhmm>***[.***<ss>***]]**

where:

<yyddmmhhmm>	Specifies the year (*yy*), month (*mm*), day (*dd)*, hour (*hh*), and minute (*mm*). If you want to set the time, but not the date, specify *<hhmm>***[.***<ss>***]**. (Actually, the software reads this argument from right to left, so you can specify any additional arguments with *<hhmm>*. For example, you can specify *<DDhhmm>* to also specify the day. Note that you must specify the arguments in the order shown. For example, you cannot enter *<YYhhmm>* or *<DDMMYYhhmm>*.)
.*<ss>*	Optionally specifies the seconds. If you use this argument, make sure to use the period (**.**) in front of the seconds. If you choose not to specify the number of seconds, the value is set to 00. If you do not use either argument, the current system date and time are displayed.

Here are some examples of the use of this command. In the first example, the current system time and date are displayed. In the second example, the time and date are changed. The new time and date are then displayed.

```
73:PowerHub:mgmt# date
wed  03/01/95  16:02:00

74:PowerHub:mgmt# date 9503310800
date set to: fri  03/31/95  08:00:00

75:PowerHub:mgmt# date
fri  03/31/95  08:00:09
```

9.3.5 Displaying the MAC-Layer Hardware Address

Each PowerHub Packet Engine has a unique MAC-layer hardware address. This address is assigned at the factory and identifies the Packet Engine, and by extension the chassis containing the Packet Engine. A label on the PowerHub 6000 lists the MAC address, and the MAC address is displayed in the boot messages. Alternatively, you can display the MAC address on the management terminal using the **ethaddr** (or **ea**) command.

Here is an example of the display produced by this command:

```
89:PowerHub:mgmt# ethaddr
Ethernet address: 00-00-ef-01-db-40
```

9.3.6 Displaying the System Configuration

To display information about the physical configuration of the PowerHub 6000, issue the **showcfg** command. Here is an example of the display produced by this command. In this example, the PowerHub 6000 contains the Packet Engine, but it does not contain a daughter card, nor does it contain any NIMs.

```
8:PowerHub:mgmt# showcfg

Accelerator board is present. Accelerator IOP is being used.
Installed DRAM Size: 16 MB
tty1:  not set - using 9600 baud
tty2:  not set - using 1200 baud
PE:    slot 1
PM:    Good
  01/01 UTP     UTP     UTP     UTP     UTP     UTP
        UTP     UTP     UTP     UTP     UTP     UTP
```

The **showcfg** command displays the following information.

- Whether the Packet Accelerator has been installed on the Packet Engine.
- The amount of DRAM installed on the Packet Engine.
- The baud rates for the TTY (RS-232) ports, shown by `tty1:` and `tty2:`. In this example, port `TTY1` is using its default baud rate 9600 because no baud rate has been explicitly set in NVRAM for that port. The `TTY2` is using its default baud rate of 1200. As noted in Section 2.3.2 on page 26, the `TTY2` port is not accessible from the Packet Engine front panel, and should be used only if recommended by FORE Systems TAC.
- The Packet Engine (shown by `PE:`) is present in slot 1.
- The status of the power supply (or supplies), shown by `PM:`. In this example, the power supply (or supplies) are functioning normally. If a problem occurs in a power supply, the following message is displayed instead of `Good`:

  `Bad: Check Power Supply LEDS on the back of the PowerHub Chassis.`

- The starting segment number in each slot in the chassis. In this example, only slot 1 (the 12 UTP segment positions on the Packet Engine) contains segments. Slots 2 through 4 are unoccupied, so they are not displayed. The segments are numbered sequentially, beginning with 1. Remember that segments are numbered from left to right, bottom to top (see Section 1.3 on page 17). The medium type in use in each segment position also is shown. The labels used for the media types are described in Table 9–2.

TABLE 9–2 Media-type labels.

Label	Medium Type
`10FB`	10Base-FB.
`10FL`	10Base-FL. (Note: In some software versions, 10Base-FL is shown as `FIBER`.)
`10T`	10Base-T. (Note: In some software versions, 10Base-T is shown as `UTP`.)
`100FX`	100Base-FX.
`100T4`	100Base-T4.
`100TX`	100Base-TX.
`AUI`	AUI.
`BNC`	BNC, unterminated.
`BNCT`	BNC, terminated.
`FIBER` (or `10FL`)	10Base-FL.

TABLE 9–2 (Continued) Media-type labels.

Label	Medium Type
MAU-FDX	MAU, in full-duplex mode (SQE heartbeat disabled).
MAU-RPT	MAU, in repeater mode (in half-duplex with SQE heartbeat disabled).
MAU-SQE	MAU, with SQE heartbeat enabled (in half-duplex mode).
MM/MM	FDDI with multimode A and B ports.
MM/SM	FDDI with multimode A port and single-mode B port.
NULL	No FORE Systems EMA, AUI Media Cable, or other FORE Systems segment connection is present. With the NULL segment type, the segment hardware and software are actually configured the same as an AUI Media Cable, with one exception: in the configuration it shows up as "NULL" rather than one of the media types. This allows you to run internal loopback tests even when no connection is present. See Section A.1 on page 220 for information about the internal loopback test.
SM/MM	FDDI with single-mode A port and multimode B port.
SM/SM	FDDI with single-mode A and B ports.
UTP (or 10T)	10Base-T.
????	No segment connector is present.

9.4 PACKET ENGINE AND NIM MANAGEMENT COMMANDS

The Packet Engine and NIM management commands let you perform the following tasks:

- Display ID and power requirements for the Packet Engine or a NIM (**`idprom`** command).
- Display the current temperature of the Packet Engine (**`temperature`** command).
- Configure the traffic LEDs on the 10 Mb/s and 100 Mb/s Ethernet segments (**`led-config`** command).

9.4.1 Displaying ID and Power Information

The Packet Engine and all types of NIMs contain a special PROM called the ID PROM. The *ID PROM* contains identification information and power requirements for the module. (Note that the UMM does not contain an ID PROM.)

You can display this information using the **`idprom`** command. Here is the syntax for this command:

`idprom|idp` *`<slot-num>`*

where:

`<slot-num>`	Specifies the slot containing the module.

Here is an example of the display produced by this command. In this example, information is displayed about the module in slot 1.

```
14:PowerHub:mgmt# idprom 1
Card Type: Packet Engine
   Serial #: 9442AL0094
      Model: 6201-00
   Revision: A
      Issue: 1
  Deviation:

  Power Requirements:
    4000 mA at 5V
```

The ID PROM display shows the following information:

`Serial #:`	The factory-assigned serial number.
`Model:`	The factory-assigned model number.
`Revision:`	The factory-assigned revision (REV) number.
`Issue:`	The factory-assigned issue number.

`Deviation:`	If applicable, the factory-assigned deviation number. Only some modules have a deviation number.
`Power Requirements:`	The maximum number of milliamps required by the module at +12-volts, +5-volts, or +3.3-volts, as applicable.

As noted above, the UMM does not contain an ID PROM. If you issue the **`mgmt idprom`** command against a slot that does not contain the Packet Engine or a NIM, the system displays the following message:

```
cannot read IDPROM information from that slot
```

This message also is displayed if you issue the command against an empty chassis slot.

9.4.2 Displaying the Temperature of the Packet Engine

As described in Section 2.3.5 on page 27, the PowerHub 6000 contains a temperature sensor that reads the temperature of the inside of the chassis with an accuracy of plus or minus 0.5° C. You can display the temperature by issuing the following command:

`temperature|temp` *`<slot-num>`***`|all`**

where:

`<slot-num>`	Specifies the NIM slot that contains the module for which you want to display the temperature.
`all`	Displays the temperature for all modules.

Note that the PowerHub 6000 is designed for operation over a range of external ambient temperatures. An additional temperature rise inside the chassis is accounted for in the design of the product.

9.4.3 Configuring the Ethernet Traffic LEDs

Using the **`led-config`** command, you can configure the traffic LEDs (C/X and A/R) on Ethernet segments. The LEDs are set as a group for an entire module, not individually on a segment-by-segment basis. You can set the LEDs on a module as C and A or as X and R. (For a description of the information indicated by each setting, see Section 2.3.11 on page 29.) The default setting is C and A. The configuration of the LNK LED cannot be changed.

Here is the syntax for the **`led-config`** command:

`led-config|lc [`*`<slot-num>`***` [ca | xr]]`**

where:

`<slot-num>`	Is the slot that contains the LEDs you want to configure. To configure the Packet Engine (and Fast Ethernet daughter card) traffic LEDs, specify slot number 1. To specify the traffic LEDs on an Ethernet NIM, specify the slot number that contains the NIM (3 or 4). If you specify a slot number, but not an LED setting, the current setting for the specified slot is shown. If you do not specify a slot number, the current LED settings for all the configurable traffic LEDs in the chassis are shown.

ca|xr Configures the LEDs:

ca Reflects transmit collision and activity (transmit and receive). The default is **ca**.

xr Reflects packet transmit and receive activity.

Here are some examples of the use of this command. In the first example, the current setting for the Packet Engine and Fast Ethernet daughter card (if present) is displayed. Although the command and system response refer only to slot 1, information for slot 2 is always included when the request is for slot 1.

```
30:PowerHub:mgmt# led-config 1
slot 01: ca (leds reflect collision and activity)
```

In the following example, the traffic LEDs on the Packet Engine (and Fast Ethernet daughter card, if present) are configured to reflect receive and transmit activity.

```
31:PowerHub:mgmt# led-config 1 xr
slot 01: xr (leds reflect transmit and receive activity)
```

9.5 SEGMENT MANAGEMENT COMMANDS

The segment management commands configure or display information about individual segments. You can use these commands to perform the following tasks:

- Assign a name to a segment (**set-portname** command).
- In systems containing the UMM (Universal Media Module), activate a UTP connector bypassed by the UMM (**activate-utp** command).
- In systems containing the UMM (Universal Media Module), re-activate an EMA or AUI Media Cable in the UMM (**activate-ema** command).
- Enable or disable automatic segment-state detection for a segment (**autoportstate** command).
- List how many times the automatic segment-state detection feature has detected that a particular segment is down or up and marked it as down or up in the PowerHub software (**port-aps-down-count** and **port-aps-up-count** commands).
- List how many times you have enabled or disabled a particular segment (**port-mgmt-down-count** and **port-mgmt-up-count** commands).
- Change the operating mode (half-duplex or full-duplex) of an Ethernet segment (**operating-mode** command).

9.5.1 Setting a Segment Name

Use the **set-portname** command to assign a name to a segment. The segment name is used in tables that contain information that relates to the segment. You can specify an alphanumeric string up to 23 characters in length. The string can include underscores. The segment name cannot contain embedded blanks.

Here is the syntax for this command:

set-portname|spn *<seg> <name>*

where:

<seg>	Specifies the segment number.
<name>	Specifies the name you are assigning to the segment.

Here are some examples of this command. The commands in this example assign descriptive names to segments 1 through 6. These names appear in place of the default names “port 1,” “port 2,” and so on in table displays.

```
81:PowerHub:mgmt# set-portname 1 Lab1
Port 1 named: Lab1

82:PowerHub:mgmt# set-portname 2 Lab2
Port 2 named: Lab2

83:PowerHub:mgmt# set-portname 3 Lab3
Port 3 named: Lab3

84:PowerHub:mgmt# set-portname 4 Lab4
Port 4 named: Lab4

85:PowerHub:mgmt# set-portname 5 Lab5
Port 5 named: Lab5

86:PowerHub:mgmt# set-portname 6 Lab6
Port 6 named: Lab6
```

9.5.2 Activating a UTP Connector

This section applies only to systems that contain the UMM (Universal Media Module). If your PowerHub chassis contains the UMM, the EMA (Ethernet Media Adapter) or AUI Media Cable for each segment (1-6) is activated by default. However, if you need to activate any of the six UTP connectors bypassed by the UMM, issue the following command:

activate-utp|au *<seg-list>*

where:

<seg-list>	Specifies the segments for which you are activating the corresponding UTP connector. You can specify a single segment, a comma-separated list, or a hyphen-separated range. Only segment numbers 1 through 6 are valid.

When you activate the UTP connector, the corresponding EMA or AUI Media Cable is deactivated and no longer receives or transmits traffic. The green Status (S) LED to the left of the EMA or AUI Media Cable goes dark.

To reactivate an EMA or AUI Media Cable, use the **activate-utp** command, described in Section 9.5.3 on page 167

9.5.3 Activating an EMA or AUI Media Cable

If your system contains the UMM, and you previously activated a UTP connector (thereby de-activating the corresponding EMA or AUI Media Cable), you can re-activate it using the following command:

activate-ema|ae *<seg-list>*

where:

<seg-list>	Specifies the segments for which you are activating the corresponding EMAs or AUI Media Cables. You can specify a single segment, a comma-separated list, or a hyphen-separated range. Only segment numbers 1 through 6 are valid.

When you re-activate an EMA or AUI Media Cable, the green S (Status) LED to the left of the EMA or AUI Media Cable lights up, indicating that the EMA or cable has been selected.

9.5.4 Setting Automatic Segment-State Detection

The *automatic segment-state detection* feature can recognize that a segment is down and automatically disable bridging and routing on that segment. When this feature detects that a segment's state has changed, it disables the segment (takes the segment out of service) and marks the change in the software as appropriate. The updated segment state is displayed when you issue the **autoportstate** command without its optional arguments.

> **NOTE**: If you disable automatic segment-state detection on a segment, the segment's state is always reported as "good" and interface states are always reported as "up" in the software. For information about a segment's or interface's state, enable automatic segment-state detection for that segment.

The automatic segment-state detection feature also updates segment-state data displayed by other commands, such as the **bridge state** command. (See Chapter 2 in the *PowerHub Software Manual, V 2.6* (Rev C).)

The method used by the software to determine whether a segment is down differs depending upon the type of segment. Table 9–3 lists the methods the PowerHub software uses to determine the state of each type of segment.

TABLE 9–3 Segment-state detection methods.

Type of segment	Segment is determined to be down if…
10Base-FB	No link-test pulses are present on this segment.
10Base-FL	No link-test pulses are present on this segment.
100Base-FX	No data or idle symbols are being received on this segment.
100Base-T4	No link-test pulses are present on this segment.
100Base-TX	No data or idle symbols are being received on this segment.
AUI	No packets are received and a “loss of carrier” is detected T times over a 1-second period, where T is specified by the *<threshold>* argument on the **autoportstate** command. Note that the PowerHub 7000 does not send test packets, but relies on client and network management traffic to detect carrier loss. The state is changed to UP if at least one packet is received. When automatic segment-state detection is first enabled (for example, when the hub is booted), each AUI segment begins in the “down” state but is changed to the “up” state as soon as it receives packets.
BNC/BNCT	No packets are received and transmit-buffer errors occur over a T-second time period, where T is specified by the *<threshold>* argument on the **autoportstate** command. The state is changed to UP if at least one packet is received. As with AUI segments, when automatic segment-state detection is first enabled (for example, when the hub is booted), each BNC segment begins in the “down” state but is changed to the “up” state as soon as it receives packets.
FDDI	The attachment configuration of the segment is “isolated.”
MAU	No AUI cable carrying +12-volt current (standard for AUI) is connected to the MAU.
UTP	No link-test pulses are present on this segment.

9.5.4.1 Software Behavior When Automatic Segment-State Detection Is Disabled

When a segment is disabled, no packets are bridged or routed on that segment. This is true whether the segment is disabled by the automatic segment-state detection feature or by you, issuing the **bridge port** command. See Chapter 2 in the *PowerHub Software Manual, V 2.6* (Rev C) for further information about the **bridge port** command. Note that if you use the **bridge state** command to display state information for a segment, the state displayed in the `Management` field is “disabled” if the segment is disabled by the **bridge port** command, but remains “enabled” if the segment is disabled by automatic segment-state detection.

9.5.4.2 *Default Setting*

The default setting for the automatic segment-state detection feature differs depending upon the segment type. Table 9–4 lists the default setting for each segment type.

TABLE 9–4 Default setting for automatic segment-state detection.

Segment Type	Default
10Base-FB	Enabled
10Base-FL	Enabled
10Base-T (UTP)	Enabled
100Base-FX	Enabled
100Base-T4	Enabled
100Base-TX	Enabled
AUI	Disabled
BNC/BNCT	Disabled
FDDI	Enabled
MAU	Enabled

As shown in Table 9–4, all segment types except AUI and BNC/BNCT have automatic segment-state detection enabled by default. In general, you should leave the automatic segment-state detection feature at the factory default settings.

9.5.4.3 *Automatic Segment-State Detection Is Disabled on AUI, BNC, and BNCT*

When automatic segment-state detection is enabled, the PowerHub software does not enable AUI, BNC, and BNCT segments until the segments receive traffic. In most configurations, the AUI, BNC, and BNCT segments are connected to devices that are prepared to generate traffic. However, if you connect AUI, BNC, and BNCT segments to AUI, BNC, and BNCT segments on another PowerHub system, and the other PowerHub system has automatic segment-state detection enabled on these segments, no traffic is exchanged by the segments. Each end of the segment waits to receive traffic before becoming enabled. As a result, neither end of the segment becomes enabled and no traffic is exchanged.

If the device at the other end of the AUI, BNC, or BNCT segment is prepared to generate traffic, you can enable automatic segment-state detection on the segment. When the segment receives traffic from the other device, the PowerHub software enables the segment.

9.5.4.4 Automatic Segment-State Detection Must Remain Enabled on 10Base-T

You must leave automatic segment-state detection enabled for all 10Base-T (UTP) segments, even if you do not plan to use the segments. If automatic segment-state detection is disabled, the Ethernet controllers on the corresponding 10Base-T segments do not stop using the forwarding buffer for those segments. Instead, they fill their transmit buffers, even though no traffic needs to be forwarded. Full buffers can negatively affect packet throughput.

9.5.4.5 Unused Segments Should Be Explicitly Disabled

For AUI, BNC, and BNCT segments, heuristics are used to determine the segment state. Occasionally, electronic noise can make an AUI, BNC, or BNCT segment appear active when it is not. When this occurs, automatic segment-state detection believes the segment is active and does not disable it. Accordingly, we recommend that you explicitly disable segments when you remove them from service.

To disable a segment, issue the following command:

bridge port <*seg-list*> **dis**

where:

<*seg-list*>	Specifies the segments you want to disable.

For 10Base-T, 10Base-FB, and 10Base-FL segments, link-test pulses are used to determine the segment state. The Attachment Configuration state is used to determine the state of FDDI segments. Each of these methods is more dependable than the heuristics used for the AUI, BNC, and BNCT segments. Therefore, you do not need to worry about electronic noise causing these segment types to appear to be enabled when they actually are disabled.

9.5.4.6 Displaying the Automatic Segment-State Detection Settings

To display the current automatic segment-state detection settings, to enable the feature on a segment, or to change the threshold for AUI or BNC segments, issue the following command:

autoportstate|aps [<*seg-list*>**|all enl|dis [**<*threshold*>**]]**

where:

<*seg-list*>**|all**	Specifies the segment(s) for which you want to enable or disable automatic segment-state detection.
enl|dis	Specifies whether you are enabling or disabling automatic segment-state detection on the specified segments.

> **NOTE**: Automatic segment-state detection must remain enabled on all 10Base-T (UTP) segments. The Ethernet controllers refuse to transmit packets on any segment that does not have a “good” link status. As a result, buffers can become “stuck” on the output queue of 10Base-T segments that do not have a “good” link status. This can adversely affect the performance of the rest of the hub. By enabling automatic segment-state detection, you can prevent buffers from being enqueued on the segments, and allow any enqueued buffers to be recovered if the segments go down.

<threshold> For AUI segments, specifies the "loss-of-carrier threshold"; that is, the number of times a loss of carrier must be detected in a 1-second period for the segment to be considered down and therefore be disabled by the software.

For BNC segments, specifies the "idle period threshold"; that is, the number of seconds during which the segment must remain idle to be considered down and therefore be disabled by the software.

The default is **10** for AUI, or **5** for BNC. This argument does not apply to other media types, such as 10Base-T and 10Base-FL.

NOTE: The threshold setting applies only to AUI and BNC segments.

Here is an example of the display produced when this command is issued without any of the optional arguments. In this example, the PowerHub 6000 contains the UMM. Segments 1 through 3 in the UMM are BNC segments. Segments 10 through 12 are AUI segments.

```
59:PowerHub:mgmt# autoportstate
Automatic port-state detection of ports state:
Port 1:  enabled (currently good), idle period threshold = 5 seconds
Port 2:  enabled (currently good), idle period threshold = 5 seconds
Port 3:  enabled (currently good), idle period threshold = 5 seconds
Port 4:  enabled (currently good), loss of carrier threshold = 10
Port 5:  enabled (currently good), loss of carrier threshold = 10
Port 6:  enabled (currently good), loss of carrier threshold = 10
Port 7:  enabled (currently bad)
Port 8:  enabled (currently bad)
Port 9:  enabled (currently bad)
Port 10:  enabled (currently bad)
Port 11:  enabled (currently bad)
Port 12:  enabled (currently bad)
```

This display shows information appropriate to each segment type:

- Because BNC segments are determined to be down if they are idle for the period specified by *<threshold>* (in this case, 5 seconds), their "idle period threshold" is shown.
- Because AUI segments are determined to be down if a "loss of carrier" is detected the number of times specified by *<threshold>* in a 1-second period, the "loss of carrier" threshold (in this case, 10 seconds) is listed.

The other types of Ethernet segments are determined to be down in the absence of regular link-test pulses, or data or idle symbols. FDDI segments are down if the attachment configuration of the segment is "isolated." In either case, no threshold is shown.

9.5.5 Displaying Segment-State Statistics

At any time, you can display the following segment-state statistics for any segment:

- The number of times the automatic segment-state feature has detected that the segment is down, and therefore marked the segment as down in the software.
- The number of times the automatic segment-state feature has detected that the segment is up, and therefore marked the segment as up in the software.
- The number of times the segment has been disabled using the **bridge port** <*seg-list*> **dis** command. (See Chapter 2 in the *PowerHub Software Manual V 2.6* (Rev C).)
- The number of times the segment has been enabled using the **bridge port** <*seg-list*> **enl** command. (See Chapter 2 in the *PowerHub Software Manual V 2.6* (Rev C).)
- You can clear the counters for any of these statistics. In addition, you can display the count since the most recent system reset (reboot), or since you last cleared the counters. By default, the count since the last clear is displayed. Use the optional **-t** argument to display the count since the last system reset.

9.5.5.1 Displaying How Many Times the Segment Has Been Detected to be Down

Use the **port-aps-down-count** command to display how many times the automatic segment-state feature has detected that a particular segment is down and taken the segment out of service. Here is the syntax for this command:

port-aps-down-count|pad <*seg-list*> **show|s|clear|c [-t]**

where:

<*seg-list*>	Specifies the segment(s) for which you want to display or clear the statistic. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.
show|s|clear|c	Specifies whether you want to display the statistic or clear the counter. If you specify **clear**, the counter is reset to 0 (zero) and begins counting again.
-t	Optionally specifies that the count since the last system reset, rather than the last clear, be displayed.

Here are some examples of the use of this command:

```
59:PowerHub:mgmt# port-aps-down-count 1 show

Count of port(s) going down because of APS: (Since last stats clear)
Slot/Port     Count
--------- -------------
  01/01                1
60:PowerHub:mgmt# port-aps-down-count 1 show -t

Count of port(s) going down because of APS: (Since last system reset)
Slot/Port     Count
--------- -------------
  01/01                3
61:PowerHub:mgmt# port-aps-down-count 1 clear
Okay
```

In this example, the first command shows that the state of segment 1 has been taken out of service one time subsequent to the last time this statistic was cleared.

The second command uses the optional **-t** argument to show the statistics for the current power cycle, rather than since the last time the counter for this statistic was cleared. Notice that the count (shown under `Count`) is different.

The final command clears the counter for the statistic, resetting it to 0 (zero).

9.5.5.2 Displaying How Many Times the Segment Has Been Detected to be Up

Use the **port-aps-up-count** command to display how many times the automatic segment-state feature has detected that a particular segment is up and put the segment back into service. Here is the syntax for this command:

port-aps-up-count|pau *<seg-list>* **show|s|clear|c [-t]**

where:

<seg-list>	Specifies the segment(s) for which you want to display or clear the statistic. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.
show|s|clear|c	Specifies whether you want to display the statistic or clear the counter. If you specify **clear**, the counter is reset to 0 (zero) and begins counting again.
-t	Optionally specifies that the count since the last system reset, rather than the last clear, be displayed.

The output produced by this command is similar to the **port-aps-down-count** output. (See the examples in Section 9.5.5.1.)

9.5.5.3 Displaying How Many Times the Segment Has Been Manually Disabled

Use the **`port-mgmt-down-count`** command to display how many times the segment has been disabled using the **`bridge port dis`** command. (See Chapter 2 in the *PowerHub Software Manual V 2.6* (Rev C).) Here is the syntax for this command:

`port-mgmt-down-count|pmd` *`<seg-list>`* **`show|s|clear|c [-t]`**

where:

`<seg-list>`	Specifies the segment(s) for which you want to display or clear the statistic. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.
`show|s|clear|c`	Specifies whether you want to display the statistic or clear the counter. If you specify **`clear`**, the counter is reset to 0 (zero) and begins counting again.
`-t`	Optionally specifies that the count since the last system reset, rather than the last clear, be displayed.

Here is an example of the use of this command:

```
59:PowerHub:mgmt#  port-mgmt-down-count 1-4 show

Count of port(s) going down because of user command: (Since last stats clear)
Slot/Port     Count       Slot/Port     Count       Slot/Port     Count
--------- -------------  --------- -------------  --------- -------------
  01/01              0    01/02              2    01/03              0
  01/04              3
```

In this example, statistics are displayed for segments 1 through 4. Because the optional **`-t`** argument was not used, the statistics shown are those collected after the last time the statistic counter was cleared. Segment 2 and segment 4 have been explicitly taken down by the **`bridge port dis`** command subsequent to the last time the counter was cleared. Segment 2 has been taken down twice. Segment 4 has been taken down three times.

9.5.5.4 Displaying How Many Times the Segment Has Been Manually Enabled

Use the **`port-mgmt-up-count`** command to display how many times the segment has been enabled using the **`bridge port enl`** command. (See Chapter 2 in the *PowerHub Software Manual V 2.6* (Rev C).) Here is the syntax for this command:

`port-mgmt-up-count|pmu` *`<seg-list>`* **`show|s|clear|c [-t]`**

where:

`<seg-list>`	Specifies the segment(s) for which you want to display or clear the statistic. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments.
`show|s|clear|c`	Specifies whether you want to display the statistic or clear the counter. If you specify **`clear`**, the counter is reset to 0 (zero) and begins counting again.

-t	Optionally specifies that the count since the last system reset, rather than the last clear, be displayed.

The output produced by this command is similar to the output produced by the **port-mgmt-down-count** command. (See the examples in Section 9.5.5.3.)

9.5.6 Changing the Operating Mode of an Ethernet Segment

You can change the operating mode for the following types of Ethernet segments using the **operating-mode** command:

- 10Base-FL
- 10Base-T
- 100Base-FX
- 100Base-TX

Here is the syntax for this command:

operating-mode|om [*<seg-list>***|all [fdx|normal]]**

where:

*<seg-list>***|all**	Specifies a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. If you specify **all**, the operating mode for all private UTP, 10Base-FL, 100Base-TX, and 100Base-FX segments is displayed or changed.
fdx|normal	Specifies whether you are placing the segment(s) into full-duplex mode (**fdx**) or half-duplex mode (**normal**).

If you leave out the **fdx|normal** argument, the operating mode for the specified segments is displayed, rather than changed. If this command is issued with no arguments, the operating mode for all segments capable of full-duplex operation is displayed.

> **NOTE**: The remote connection must be compatible with the PowerHub implementation of full-duplex operation. Because full-duplex operation has not become fully standardized yet, the PowerHub full-duplex mode might not be compatible with some third-party devices supporting full-duplex mode. Full-duplex mode is always compatible among like 10Base-FL, 10Base-T, 100Base-TX, and 100Base-FX Fast Ethernet segments in the PowerHub 6000. The PowerHub 6000 accomplishes full-duplex operation simply by disabling collision detection.

9.6 FILE MANAGEMENT COMMANDS

The Flash Memory Module uses a DOS-based file system that provides file services to the Packet Engine. For example, when you read or save an environment file or configuration file, the DOS-based file system is used to read or write the appropriate file.

You can use this file system to perform the following tasks:

- Display a directory of files (**dir** command).
- Display volume information as well as a directory of files (**listdir** command).
- Display a file on the management terminal (**showfile** command).
- Copy a file (**copy** command).
- Rename a file (**rename** command).
- Remove a file (**fremove** or **remove** command).
- Generate a checksum for verifying the integrity of a file (**checksum** command).
- Reformat the Flash Memory Module (**format** command).

9.6.1 File Naming Conventions

File names conform to the following rules:

- The maximum length is eight characters, optionally followed by an extension from one to three characters in length (ex: `filename.ext`).
- File names are not case sensitive. For example, the file names `filename` and `FILENAME` are considered to be identical by the file system.
- You can use any combination of letters, numbers, or special characters except the following in a file name: `*`, `-`, `!`, `?`, `:` .
- You can use the following wildcards in place of any portion of a file name:

  ***** Allows any character string. For example, the file specification `*.env` yields all file names that end with the extension “env.”

  ? Allows any single character. For example, the file specification `hub?.cfg` yields all file names that begin with “hub,” contain one additional character following “hub,” and have the extension “cfg.”

  You can combine both wildcards in the same file specification. For example, `hub?.*` is a valid file specification.

9.6.2 Displaying Directory and Volume Information

Use the **`listdir`** command to display a directory of the Flash Memory Module. Here is the syntax for this command:

`listdir|ls [`*`<filespec>`***`]`**

`where:`

`[`*`<filespec>`***`]`**	Specifies a file name. You can use the wildcards (**`*`** and **`?`**) for any portion of the file name.

This command is designed to present directory information in a way familiar to UNIX users. Here is an example of the type of display produced by the **`listdir`** command:

File Name

File Size

```
61:PowerHub:mgmt# listdir

      3116    5-17-1995   3:50p FM:FORE.DMP
      3322   12-15-1994  11:49a FM:INTLOOP
      2807   12-15-1994  11:49a FM:EXTLOOP
    902117    5-05-1995   2:18p FM:6PE
       226    4-26-1995   1:00p FM:ROOT.ENV
        27    5-17-1995   2:59p FM:BOOTDEF
      1622    5-17-1995   3:07p FM:LAB.1
      1000    7-13-1995   3:07p FM:HAVERTY.PAT
```

Date that file was written or changed on the device

Device on which file is stored

Time that file was written or changed on the device

The “`FM:`” in front of each file name merely indicates that the file is on the Flash Memory Module.

> **NOTE**: The PowerHub 6000’s DOS-like file system does not support hierarchical directories.

Use the **`dir`** command to display a file directory that includes volume information and available space for the storage device. Here is the syntax for this command:

`dir [`*`<filespec>`***`]`**

where:

`<filespec>`	Specifies a file name. You can use the wildcards (**`*`** and **`?`**) for any portion of the file name.

Whereas the **`listdir`** command presents a UNIX-like directory, the **`dir`** command presents a DOS-like directory. Here is an example of the display produced by the **`dir`** command:

File Size

File Name

```
62:PowerHub:mgmt# dir

Volume in device is 6-2.6.3.0

FORE      DMP        3116    5-17-1995   3:50p
INTLOOP              3322   12-15-1994  11:49a
EXTLOOP              2807   12-15-1994  11:49a
6PE                902117    5-05-1995   2:18p
ROOT      ENV         226    4-26-1995   1:00p
BOOTDEF                27    5-17-1995   2:59p
LAB.1                1622    5-17-1995   3:07p
HAVERTY.PAT          1000    7-13-1995   3:07p

      8 File(s)      540760 bytes free
```

Date that file was written
or changed on the device

Time that file was written
or changed on the device

9.6.3 Displaying a File

Use the **`showfile`** command to display a file contained in the Flash Memory Module. The file's contents are written to the management terminal. Here is the syntax for this command:

`showfile|shf` *`<file-name>`*

where:

`<file-name>` Specifies the name of the file you want to display.

> **NOTE**: You can control the number of lines displayed at one time using the **`main stty`** command. (See Section 8.7 on page 143 for information.)

9.6.4 Copying a File

Use the **`copy`** command to copy a file contained in the Flash Memory Module. Here is the syntax for this command:

`copy|cp` *`<src-file>`* *`<dest-file>`*

where:

`<src-file>`	Is the name of the file you are copying.
`<dest-file>`	Is the name of the new file into which you are copying the contents of the *`<src-file>`*.

> **CAUTION**: If the destination file you specify is an existing file, the **`copy`** command overwrites that file. No warning message is displayed.

If the storage medium (Flash Memory Module or floppy diskette) contains enough free space to hold the copy, the file is copied. If the storage medium does not contain enough free space for the copy, the following message is displayed: `device is full`.

9.6.5 Renaming a File

Use the **`rename`** command to rename a file contained in the Flash Memory Module. Here is the syntax for this command:

`rename|ren` *`<oldfilename>`* *`<newfilename>`*

where:

`<oldfilename>`	Is the name of the file you are renaming.
`<newfilename>`	Is the new name for the file.

If you specify the name of an existing file for the new file name, the software does not overwrite the existing file, but instead displays the following error message: `file exists`.

9.6.6 Removing a File

Use the **`fremove`** command or the **`remove`** command to remove a file from the Flash Memory Module. Both commands do the same thing.

> **NOTE**: The **`fremove`** and **`remove`** commands prompt you to enter **`y`** or **`n`** to confirm removal of a file. There is no “unremove” command, so be careful! Also, the software image files are not protected. They can be removed by PowerHub commands.

Here is the syntax for this command:

fremove|frm [-f] [-i] *<filespec>* **[***<filespec>...***]**

where:

-f	Forces the software to remove the file(s), without presenting a warning prompt.
-i	Overrides the **-f** (Force) flag, presenting a warning prompt before removing each file. The prompt gives you the opportunity to cancel your request to remove the file. The default is **-i**.
<filespec>	Specifies the name of the file you want to delete.

The syntax for the **remove** command is the same.

Here is an example of the **fremove** command:

```
63:PowerHub:mgmt# fremove bridge.cfg
remove FM:BRIDGE.CFG? y
frm: FM:BRIDGE.CFG: deleted
```

9.6.7 Calculating a Checksum

Use the **checksum** command to generate a checksum of a file in the Flash Memory Module. The checksum always produces the same result for identical files. Therefore, you can use the checksum to verify the integrity of duplicate files or files transmitted from one PowerHub Intelligent Switching Hub to another over the network.

For example, suppose you want to copy an environment file from one hub to another over the network. Before copying the file, use the **checksum** command to calculate the checksum for the file. Then use the **tftp put** command to write the file to a TFTP server. From a command-line session on the target hub, download the file onto that hub using the **tftp get** command. (For information on the TFTP commands, see Chapter 4 in the *PowerHub Software Manual, V 2.6* (Rev C).)

To ensure that the file has arrived intact, you can issue the **checksum** command against the copy of the file received on the target hub. If the checksums match, the file very likely is intact. However, if the checksums do not match, an error must have occurred during transmission or storage.

Here is the syntax for the **checksum** command:

checksum|sum *<filespec>* **[***<filespec>...***]**

where:

<filespec>	Specifies a file name. You can use the wildcards (***** and **?**) for any portion of the file name.
<filespec>...	Specifies additional files. To calculate individual checksums for all the files on the Flash Memory Module, enter **checksum *** .

Here is an example of the use of this command:

```
67:PowerHub:mgmt# checksum hub1.cfg
0x0fcf9636 HUB1.CFG
```

The software calculates the checksum for the specified file(s) and displays the checksum(s).

9.6.8 Reformatting the Flash Memory Module

Generally, you can prevent filling up your Flash Memory Module by removing files you no longer need. The module contains more than enough space to hold the system software and self-test scripts, as well as your configuration files, environment files, and dump file(s). However, you can use the **`format`** command to reformat the Flash Memory Module. When you reformat the module, all the system software and data files are physically removed from the module.

> **WARNING**: This command does not display a confirmation message. The software begins reformatting the Flash Memory Module as soon as you issue the command. Before you reformat the Flash Memory Module, make sure you copy any files you want to keep onto another device, such as a floppy diskette or a file server.

Here is the syntax for this command:

`format|fmt` *`<device-name>`*

where:

`<device-name>`	Specifies the file device. Specify **`fm`** as the device name.

9.7 TTY PORT MANAGEMENT COMMANDS

In addition to the RS-232 session through TTY1, you can have, additional, simultaneous command-line sessions on the hub.

The sessions differ according to the connection type:

TTY	Asynchronous link from your workstation or PC to the TTY1 or TTY2 port.
TELNET	Network (IP) link from your network node (workstation or PC) to the hub.

The command-line interface looks and behaves the same whether the session is on a TTY (RS-232) port or a TELNET connection. A standard terminal with 80-character lines is assumed. The interface does not use the cursor-positioning, screen-clearing, or other display features of VT100 or other “smart” terminals.

The TTY management commands let you perform the following tasks:

- Set the baud rate on TTY1 or TTY2 (**setbaud** command)
- Open a session on the TTY2 port (**tty2open** command).
- Close the session on the TTY2 port (**tty2close** command).

> **NOTE**: The TTY2 port is not accessible from the Packet Engine front panel. This port is used for diagnostic purposes. Use this port only if advised to do so by FORE Systems TAC.

9.7.1 Setting the Baud Rate for a TTY Port

You can set the baud rate (data transmission rate) for the TTY ports using the following command:

setbaud|sb tty1|tty2 *<baud-rate>*

where:

tty1|tty2	Specifies the port for which you are setting the baud rate.
<baud-rate>	Specifies one of the following baud rates: **1200**, **2400**, **4800**, **9600**, or **19200**. The default is **9600** for TTY1; **1200** for TTY2.

The newly-specified baud rate is stored in NVRAM and takes effect immediately. It is retained across logins and across power cycles.

> **NOTES**: If the Lock Switch is unlocked when you boot the hub, the TTY ports use their default baud rates (9600 for TTY1 and 1200 for TTY2), regardless of the baud rates stored in NVRAM.
>
> Before you can set the baud rate for TTY2, a session must be opened on the port. To open a TTY2 session, issue the **tty2open** command.

Here is an example of the use of this command:

```
97:PoweHub:mgmt# setbaud tty2 19200
Changed tty2 baud rate to 19200; written to nvram
```

9.7.2 Opening a Session on the TTY2 Port

If advised to do so by FORE Systems TAC, you can attach a management terminal or modem to the TTY2 port and open a management session on that port. Use the **tty2open** command to open the session.

As with other command-line sessions, if the Lock Switch (or jumper) is Locked, you must enter a login (`root` or `monitor`) and the corresponding password before the command prompt is displayed.

9.7.3 Closing the Session on the TTY2 Port

To close the TTY2 session, issue this command from the command prompt in a TTY1 or TELNET session: **`tty2close`**.

> **NOTE**: No warning message is displayed for the user on the TTY2 session. Closing the TTY2 session is equivalent to logging the user out of the session on TTY2.

9.8 CONFIGURATION FILE COMMANDS

A configuration file is an ASCII file containing PowerHub commands that set configuration parameters and definitions including:

- Segment enable or disable.
- Network (bridge) groups and filters.
- Spanning-tree enable or disable.
- Protocol interfaces and filters.
- TELNET control characters.
- PowerHub system name and location description.
- Segment names.

Normally, when you power down the PowerHub 6000 or reboot the software, any changes made to these configuration items are lost. However, if you save a configuration file before powering down or rebooting, commands to create or set the current configuration items are placed into an ASCII file. If you read (load) that file at boot time or during a command-line session, the commands in the file recreate the saved configuration items and settings.

Note that configuration parameters, unlike environment parameters, affect the entire hub, rather than just the current command-line session. Consequently, if you read a configuration file during a command-line session, configuration changes caused by the commands in the file have global effect. Users on other command-line sessions are affected by the changes.

Just as with environment files, you can save or load multiple configuration files during a command-line session; also like environment files, the effects of multiple configuration files are cumulative.

You can load a configuration file automatically during power on or a reboot. When you boot the software, the software checks for a file named `cfg`, the default configuration file name. If this file is present, the software assumes it is a configuration file and loads the file. The commands in the file are executed to configure the hub. If the hub is booted from a TFTP server, the software looks for the configuration file name specified in the hub's `bootdef` file on the boot server. (See Section B.4 on page 238.)

Whenever you make configuration changes that you want to add to the default configuration file, you can add those changes by saving the configuration file.

- If the configuration file is located on the Flash Memory Module, issue the **`savecfg cfg`** command.
- If the configuration file is located on a TFTP server, save it by issuing the **`tftp svcfg`** *`<remfile>`* command.
- If you use both the Flash Memory Module and a TFTP server as boot sources, issue both commands.

See Section 9.8.2 for information on these commands.

> **NOTE**: If you are booting from a TFTP server ("netbooting"), you need to add a line to the boot definition file on the boot server that identifies the directory location and name of the configuration file; the `cfg` file is not read from the Flash Memory Module when you netboot.

9.8.1 Example Configuration File

Example 9–1 shows a PowerHub configuration file. Notice that the file contains comments (preceded by #) and PowerHub commands. Configuration files are organized according to subsystem. The commands shown under each subsystem are ordinary PowerHub commands. You could issue these commands manually (and in the same order) to achieve the same configuration this file achieves. Configuration files simply make using the PowerHub 6000 more practical by issuing the commands automatically when the system is booted.

In every configuration file, the last command is **`endcfg`**. This command must be present at the end of the configuration file. If you edit a configuration file using an ASCII editor, make sure you add this command to the end of the file.

```
#########################################################################
#
#        Main Configuration
#
main
#########################################################################
#
#        Bridge Configuration
#
bridge
#########################################################################
#
#        Bridge Configuration (filter templates)
#
#########################################################################
#
#        Bridge Configuration (logical filtering rules)
#
#########################################################################
#
#        Bridge Configuration (spanning tree)
#
set aging 60
set st bridge-priority 8000
set st timer-threshold 21 4 16
set st port-priority 1,2,3,4,5,6,7,8 80 80 80 80 80 80 80 80
set st port-priority 9,10,11,12 80 80 80 80
st dis
#########################################################################
#
#        Bridge Configuration (port configuration)
#
po 1 enl
po 2 enl
...remaining segment enable commands omitted for brevity.
#########################################################################
#
#        Bridge Configuration (groups)
#
set group del default
set group all default
#########################################################################
#
#        Bridge Configuration (node table)
#
#########################################################################
#
#        Bridge Configuration (IPX translation table)
#
ibt dis
```

EXAMPLE 9–1 PowerHub configuration file.

```
#########################################################################
#
#        Mgmt Configuration
#
mgmt
dcd dis
spn 1 Port_1
spn 2 Port_2
spn 3 Port_3
spn 4 Port_4
spn 5 Port_5
spn 6 Port_6
spn 7 Port_7
spn 8 Port_8
spn 9 Port_9
spn 10 Port_10
spn 11 Port_11
spn 12 Port_12
sn PowerHub
sl Undefined
aps 1 enl
aps 2 enl
aps 3 enl
aps 4 enl
aps 5 enl
aps 6 enl
aps 7 enl
aps 8 enl
aps 9 enl
aps 10 enl
aps 11 enl
aps 12 enl
#########################################################################
#
#        Diag Configuration
#
diag
rs rdis
rs xenl
brk dis
#########################################################################
#
#        FDDI parameters
#
fddi
#########################################################################
#
#        IP interface configuration
#
ip
```

EXAMPLE 9–1 PowerHub configuration file.

```
##########################################################################
#
#        IP Routing table
#
ip
##########################################################################
#
#        IP address trnslation table
#
ip
saa 5

##########################################################################
#
#        SNMP Communities Table
#
snmp
add c public ro
##########################################################################
#
#        TCP connection timers
#
tcpstack
se ci 20
se ka 75
##########################################################################
#
#        TELNET control characters
#
tcpstack
scd ec ^H
scd el ^U
scd ew ^W
scd i ^C
scd rl ^R
scd xoff ^S
scd xon ^Q
##########################################################################
#
#        RIP configuration
#
rip
rip-ctrl-type normal
set 1 ta,li,rs,rd,ad yes
set 2 ta,li,rs,rd,ad yes
...additional RIP settings omitted for brevity.
set 1 po no
set 2 po no
...additional RIP settings omitted for brevity.
ip ahp 37
ip ahp 42
ip ahp 53
ip ahp 67
ip ahp 68
ip ahp 69
ip ahp 98
ip ahp 137
ip ahp 138
```

EXAMPLE 9–1 PowerHub configuration file.

```
##########################################################################
#
#        DECnet configuration
#
##########################################################################
#
#        IPX configuration
#
##########################################################################
#
#        AppleTalk configuration
#
##########################################################################
#
#        IP Multicast Routing Parameters
#
ipm
se imf dis
se po 1 enl
se po 2 enl
se po 3 enl
se po 4 enl
se po 5 enl
se po 6 enl
se po 7 enl
se po 8 enl
se po 9 enl
se po 10 enl
se po 11 enl
se po 12 enl
##########################################################################
#
#        End config
#
endcfg
```

EXAMPLE 9–1 PowerHub configuration file.

9.8.2 Saving Configuration Changes

You can save configuration changes to the `cfg` file (or the hub's configuration file on the TFTP server), in which case they are automatically applied each time you reboot the software. Alternatively, you can save the changes to a different file name or to a device attached to TTY1 or TTY2. If you save configuration changes to a file other than `cfg`, you must load the file after the hub is booted to apply the configuration settings in the file to the hub.

9.8.2.1 Saving the Configuration to the Flash Memory Module

To save a configuration file to the Flash Memory Module, issue the following command:

savecfg|svcfg *<file-or-device-name>*

where:

<file-or-device-name>	Specifies a file name or one of the TTY ports (TTY1 or TTY2).

Here are some examples of the use of this command. In the following example, the current configuration is saved to a file named `Lab1.cfg`.

```
99:PowerHub:mgmt# savecfg Lab1.cfg
Configuration saved to Lab1.cfg
```

The following example shows how the current configuration is saved to port TTY2.

```
100:PowerHub:mgmt# savecfg tty2
Configuration saved to tty2
```

This command simply prints the configuration file to port TTY2. This process might take a while to complete.

9.8.2.2 Saving the Configuration to a TFTP Server

To save a configuration file to a TFTP server, issue the following command:

tftp svcfg [-h *<host>***]** *<remfile>*

where:

-h *<host>*	Specifies the IP address of the TFTP server. Unless you have already specified a default TFTP server using the **tftp set** command, you need to include this argument. For information on the **tftp set** command, see Chapter 4 in the *PowerHub Software Manual, V 2.6* (Rev C).
<remfile>	Specifies the configuration file name. Specify a name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `fore` and this directory is specified as the TFTP home directory, do not specify `fore` as part of the file name.

NOTES: Some TFTP servers require that the remote file name exist on the server before you can write to that file name. If your server requires that the file name already exist, create a zero-length file (named the same as your configuration file) on the server, then specify that file name for *<remfile>*.

On some TFTP servers, including servers running Sun/OS 4.x, files that you overwrite on the server are not properly truncated. When you overwrite an existing file on the TFTP server, if the older version of the file is longer than the new file, the older version is not truncated properly by the server. As a result, the new version of the file contains part of the older version of the file.

Here is an example of the use of this command.

```
99:PowerHub:mgmt# tftp svcfg 147.128.128.7 Lab1.cfg
Configuration saved to Lab1.cfg
```

9.8.3 Reading (Loading) a Configuration File

When you boot the software, the system looks for a configuration file:

- If you boot from the Flash Memory Module, the system looks in the boot definition file on the Flash Memory Module for the configuration file name. The default configuration file name is `cfg`.
- If you boot from the network, the system looks in the boot definition file on the TFTP server for the configuration file name. This name is manually specified by you or another system administrator when you configure the PowerHub 6000 for netbooting. (See Section 4.3.4.3 on page 62 and Section B.4 on page 238.)

Even if the system finds and loads a configuration file when the software is booted, you can read (load) additional configuration files during a PowerHub session using the **`readcfg`** command.

> **NOTE**: The new configuration information does not undo the configuration information used from the `cfg` file. Instead, the new configuration is added to the current configuration, until the hub is powered down or rebooted.

9.8.3.1 Loading a Configuration From the Flash Memory Module

To load a configuration file located on the Flash Memory Module (or a configuration file saved on a terminal connected to the `TTY1` port), issue the following command:

`readcfg|rdcfg` *`<file-or-device-name>`*

where:

`<file-or-device-name>`	Specifies a file name or a TTY port. To specify `TTY1`, enter **`tty1`**.

9.8.3.2 Loading a Configuration From a TFTP Server

To load a configuration file located on a TFTP server, issue the following command:

`tftp rdcfg [-h` *`<host>`***`]`** *`<remfile>`*

where:

`-h` *`<host>`*	Specifies the IP address of the TFTP server. Unless you have already specified a default TFTP server using the **`tftp set`** command, you need to include this argument. For information on the **`tftp set`** command, see Chapter 4 in the *PowerHub Software Manual, V 2.6* (Rev C).
`<remfile>`	Specifies the configuration file name. Specify a name that is meaningful to the TFTP program on the server. For example, if the server contains a subdirectory called `fore` and this directory is specified as the TFTP home directory, do not specify `fore` as part of the file name.

9.8.4 Rebooting Without Loading the Default Configuration File

If you need to reboot the software without loading the configuration file, use the **`rename`** command to temporarily rename the default configuration file. For example, if the boot definition file names the configuration file `cfg`, rename `cfg` to `temp.cfg`. (See Section 9.6.5 on page 179 for information about the **`rename`** command.)

9.8.5 Editing a Configuration File

Although you can automatically write changes to a configuration file using the **`savecfg`** command, you also can manually edit the file directly using a text or ASCII editor. To move the file to a machine that contains an ASCII editor:

- If the editor is on a file server, use the **`tftp put`** command to write the file to the server, edit the file, then use the **`tftp get`** command to download the edited file back onto the Flash Memory Module.
- If the editor is on a PC or Macintosh, use the boot PROM **`zs`** command to write the file to the PC or Macintosh, edit the file, then use the **`zr`** command to transfer the edited file to the Flash Memory Module. (See Section 10.7 on page 207.)

9.8.6 Capturing Configuration Information

The system software diskettes and Flash Memory Module contain a configuration file that enables you to capture configuration information about the PowerHub system. The file is called `dispcfg`. When you read the file (using the **`mgmt readcfg dispcfg`** command), the commands in the file display configuration information on the management terminal.

If you experience problems with your PowerHub system, FORE Systems TAC might ask you to read the `dispcfg` file so they can use the information displayed on your management terminal when you read the file to help you resolve the problems with your system.

To read the `dispcfg` file, issue the following command:

```
mgmt readcfg dispcfg
```

This command reads the following configuration file.

```
main
main sy more dis

mgmt da
main sup
main ver

bridge s all all
bridge sa
bridge scf
bridge bt
bridge dc
bridge sc

ip it
ip sh
ip at
ip rt
ip scf
ip s arp
ip s ip
ip s icmp
ip dc
ip sc arp
ip sc ip
ip sc icmp

rip scf
rip s

atalk scf
atalk it
atalk dc
atalk czt
atalk cit
atalk nt
atalk rt
atalk at

ipx dc
ipx it
ipx s ipx
ipx scf
ipx st

diag bu
diag lr
diag swd

nvram show

mgmt aps
mgmt pm
mgmt idp all
mgmt om
mgmt dir
mgmt sum 6pe
mgmt sum 6f
```

EXAMPLE 9–2 Contents of the `dispcfg` configuration file.

```
mgmt da
bridge s all all
ip s arp
ip s ip
ip s icmp

tcpstack scf

snmp scf

fddi ssm all all
```

EXAMPLE 9–2 (Continued) Contents of the `dispcfg` configuration

9.9 PORT MONITORING

The Port Monitoring feature lets you connect a protocol analyzer (such as a Sniffer, LANalyzer, or Network Pharaoh) to a PowerHub segment and monitor the traffic on any other segment or set of segments on the same hub. Rather than separately attaching your traffic analyzer to each segment you want to monitor, you can attach the analyzer to one segment, then use the Port Monitoring commands to select the segments you want to monitor.

You can change which segments are being monitored without moving the protocol analyzer to another segment. In addition, you can monitor traffic on more than one segment simultaneously, without needing to use multiple protocol analyzers on multiple segments.

To use the Port Monitoring feature:

- You must be logged on with root capability.
- Forwarding must be disabled on the segment to which you attach your analyzer (the ***monitoring*** segment). To disable forwarding, issue the following command:

  `bridge port` *`<seg-list>`* **`dis`**

  where:

  `<seg-list>` Specifies the monitoring segment(s).

> **NOTE**: If you find that you frequently use the Port Monitoring feature, but do not want to disable forwarding on the monitoring segment each time you use this feature, you might want to dedicate one of the PowerHub segments solely to Port Monitoring.

9.9.1 How Port Monitoring Works

Conceptually, Port Monitoring “copies” packets from the monitored segments to the monitoring segments. Actually, packets are not really copied because this would dramatically reduce performance. Instead, a pointer to a monitored packet is placed on the transmit queue for the monitoring segment(s), and the packet buffer is freed up only after the packet has been transmitted both to its normal destination, if any, and to the monitoring segment(s).

For monitoring purposes, packets are classified into three types:

Incoming — A packet that is received on the monitored segment. An incoming packet might or might not be forwarded by the PowerHub 6000, according to the usual bridging and routing rules.

Forwarded — A packet that the PowerHub 6000 receives on one of its segments, then transmits on the monitored segment.

Generated — A packet that the PowerHub 6000 transmits on the monitored segment as required by the hub’s internal protocol stacks, including:

- Outgoing TCP packets in TELNET sessions.
- UDP packets for RIP updates and SNMP replies.
- ARP requests and replies.
- ICMP packets for various IP routing errors.
- Spanning-Tree hello and topology-change packets.
- Various packets generated by the IPX, AppleTalk, and DECnet protocol stacks.

Port Monitoring monitors packets regardless of any filters defined on the monitoring segment. This includes any filters that normally block traffic from the monitored segment to the monitoring segment. Filters defined on the monitored segment do remain in effect. In addition, incoming packets are monitored regardless of a segment’s Spanning-Tree state (blocked or forwarding) or the enabled state (enabled or disabled) of the segment.

9.9.2 Performance Considerations and Operation Notes

In general, the Port Monitoring feature does not adversely affect PowerHub performance on the monitored segments or other segments. However, if the monitored traffic load is greater than the capacity of the monitoring segment, then not all monitored packets are successfully queued. Packets not queued onto the monitoring segment for this reason are still delivered to their normal destinations.

When multiple segments are monitored, packets from all segments are queued onto the monitoring segment in the approximate order in which they were received, forwarded, or generated. You must determine which traffic belongs to which segment in the resulting, combined, outgoing stream. Note that if a packet is “incoming” on one monitored segment and “forwarded” on another monitored segment, only one copy of the packet is queued onto the monitoring segment.

When outgoing (forwarded or generated) ***and*** incoming packets are monitored on a segment, the packets might not appear on the monitoring segment in the same order as they appear on the monitored segment. This can happen because the software queues outgoing packets for transmission on the monitoring segment at the same time the software queues the packets for transmission on the monitored segment. This means that packets appear on the monitoring segment when they are queued for transmission, not when they are actually transmitted.

Therefore, it is possible for one or more packets to be received on the monitored segment and queued up ***after*** the outgoing packet on the monitoring segment, even though they appear on the monitored segment ***before*** the outgoing packet is actually transmitted. However, the order of packets within either the incoming stream or the outgoing stream on the monitored segment ***is*** preserved on the monitoring segment.

NOTES: Incoming runt packets, giant packets, and packets with FCS or frame-alignment errors are not monitored. Long (larger than 1518 bytes) packets on FDDI segments are fragmented by the hub and the fragments appear on the protocol analyzer. This applies even if the monitoring segment and monitored segment both are FDDI segments.

Do not use the monitoring segment for routing or any other purpose except monitoring. The monitoring segment should not have any devices connected to it other than a protocol analyzer. Other types of connected devices (workstations, servers, and so on) can get very confused by packets from monitored segments.

9.9.3 Packet Modifications

During normal bridging and routing, the PowerHub 6000 modifies certain packets before forwarding them. For example, it modifies both the MAC-layer and network-layer (routing) headers in routed packets. Moreover, when the hub forwards packets from FDDI to Ethernet, or vice versa, it modifies the packets accordingly.

With Port Monitoring, the ***modified*** packet, not the original packet, is transmitted to the monitoring segment. As a result, the packet displayed by your protocol analyzer is the modified packet. The way the packet is modified depends upon the segment type (Ethernet or FDDI) and the forwarding algorithm that is used, as summarized in Table 9–5.

TABLE 9–5 Packet modifications on monitoring segment.

Traffic Type	Monitored Segment Type	Monitoring Segment Type	Packet is...
Bridged Description: Forwarded, or incoming but not forwarded.	Ethernet	Ethernet	U
	Ethernet	FDDI	T
	FDDI	Ethernet	T
	FDDI	FDDI	TT/U
Routed Traffic Description: Forwarded.	Ethernet	Ethernet	M, I, R
	Ethernet	FDDI	M, I, R, T
	FDDI	Ethernet	M, I, R, T
	FDDI	FDDI	M, I, R, TT/U
Routed Traffic Description: Incoming but not forwarded.	Ethernet	Ethernet	I
	Ethernet	FDDI	I, T
	FDDI	Ethernet	I, T
	FDDI	FDDI	I, TT/U
Generated Traffic Description: Generated.	Ethernet	Ethernet	U
	Ethernet	FDDI	T
	FDDI	Ethernet	U
	FDDI	FDDI	U
KEY: I=IP TTL and checksum changed M=MAC address changed R=Routing header changed U=Unmodified T=Translated TT/U=Double Translated but Unchanged.			

The modifications made to packets appearing on the monitoring segment are further explained by the following key:

U The packet is not changed in any way. If the packet also undergoes a double-translation (denoted in Table 9–5 by TT), this means the packet is double-translated by the hub, but the resulting packet that is displayed on the protocol analyzer is identical to the packet before double translation.

M The destination MAC address is changed to the address of the next hop. The source MAC address is changed to the address of the hub.

I If the packet is an IP packet, certain fields in the IP header are changed. Specifically, the TTL field is decremented and the IP-header checksum is incremented. The IP header and payload are otherwise unmodified.

R Certain fields in the network header (for example, the IP header) might be changed, depending upon the routing protocol:

AppleTalk

The hop count is increased by one. Also, if the packet contains a checksum, the checksum is changed appropriately.

IP If the packet has an options field specifying source routing or route tracing, the appropriate modifications are made. If IP security options (RFC 1108) are used, then option fields may be added to or removed from the header. In rare cases, adding options fields causes the packet to exceed 1518 bytes, and consequently become fragmented.

IPX The only IPX field that is changed in the header is the “Transport Control” field. This field is incremented by 1 for each router that the packet passes through. (This field is similar to the TTL field in the IP header.) Note, however, that the MAC header can change in many different ways.

In the simplest case, where there is no header translation, the MAC header is changed as follows:

src-mac-addr Changed to address of the hub.

dst-mac-addr Changed to address of either the destination node or the next hop gateway.

When header translation is involved, in addition to the two fields above, the header type changes from the configured type for the receiving network to the configured type for the destination/next-hop network. The four different encapsulation types used for IPX are IEEE 802.3 (Raw), IEEE 802.2 (LLC), IEEE 802.2 (SNAP) and Ethernet-II.

DECnet

The following fields in the MAC header are changed:

`src-mac-addr`	Changed to address of this router.
`dst-mac-addr`	Changed to address of either the destination node or the next hop gateway.

In addition, a change is made to the DECnet long data packet headers (these are normal data packets). The long data header contains a "flags" field. This field is modified as follows:

- If the source and destination nodes of the packet are both on the same segment, the "INTRA ETHERNET PKT" bit is set.
- If the source and destination nodes are on different segments, the "INTRA ETHERNET PKT" bit is cleared.
- If the destination node is not reachable and the sender has set the "RETURN TO SENDER REQ" bit, the hub clears this bit and sets the "RETURNING TO SENDER" bit. In this case, the MAC header is changed as follows:

`src-mac-addr`	Changed to the PowerHub MAC-layer hardware address.
`dst-mac-addr`	Changed to the MAC-layer hardware address of the sender.

T	The packet undergoes translation between Ethernet and FDDI formats. In the case of long FDDI IP packets (larger than 1518 bytes), the packet also undergoes IP fragmentation. (Long non-IP packets are not monitored.)
TT/U	The packet undergoes a double translation, from FDDI to Ethernet format and back. The end result normally appears unchanged, except for fragmentation in the case of long IP packets. (Long non-IP packets are not monitored.)

9.9.4 Monitoring a Segment

Use the **`port-monitor`** command to monitor one or more segments. Here is the syntax for this command:

`port-monitor|pm view` *`<options>`***`|all`** *`<seg-list>`*
`on` *`<seg-list>`*

where:

*`<options>`***`|all`** Specifies which packets are to be monitored. *`<options>`* can be one or more of the following:

- **`incoming|i`** Monitors all traffic received on the segment(s) by the hub.
- **`forwarded|f`** Monitors all traffic forwarded to the segment(s) by the hub.
- **`generated|g`** Monitors all traffic generated on the segment(s) by the hub.

Separate each option with a comma. To specify all three options, use **`all`**.

`<seg-list>` Specifies which segments are to be monitored. For most applications of Port Monitoring, this segment list contains just one segment.

`on` Is a required argument that introduces the monitoring segment(s).

`<seg-list>` Indicates the monitoring segments to which the monitored traffic is to be sent. For most applications of Port Monitoring, this segment list contains just one segment.

Here are some example of the use of this command. In the first example, the command sets up segment 12 to monitor all incoming packets received on segment 10.

```
1:PowerHub:mgmt# pm view i 10 on 12
Port 10 (i) being viewed on: 12
```

The following example sets up segment 6 to monitor all packets forwarded by the PowerHub 6000 to segment 2.

```
2:PowerHub:mgmt# pm view f 2 on 6
Port 2 (f) being viewed on: 6
```

The next example sets up segment 4 to monitor all packets generated by the PowerHub 6000 on segment 13.

```
3:PowerHub:mgmt# pm view g 13 on 4
Port 13 (g) being viewed on: 4
```

The following example sets up segment 7 to monitor all packets on segment 10. This includes all incoming, forwarded, and generated packets on segment 10.

```
4:PowerHub:mgmt# pm view all 10 on 7
Port 10 (all) being viewed on: 7
```

This final example sets up segments 11 and 12 to monitor all packets that are generated on segments 5,6, and 8.

```
6:PowerHub:mgmt# pm view g 5,6,8 on 11,12
Ports 5,6,8 (g) being viewed on: 11,12
```

9.9.5 Monitoring Traffic Between a Pair of Segments

In addition to monitoring traffic on specific segments, you also can monitor traffic that the PowerHub 6000 forwards between a specific pair of segments, using this command:

port-monitor|pm viewpair *<seg1> <seg2>* **on**

<seg-list> **[r]**

where:

viewpair	Is a required argument specifying that you want to monitor traffic between a specific pair of segments.
<seg1>	Specifies which segment is to be monitored as the receiving segment.
<seg2>	Specifies the segment to which packets received on *<seg1>* are forwarded.
on	Is a required argument that introduces the monitoring segment(s).
<seg-list>	Specifies the segment(s) to which your protocol analyzer is attached.
r	Specifies that traffic received on *<seg2>* and forwarded on *<seg1>* also is monitored.

Here are some examples of the use of this command. The first example sets up segment 1 to monitor all packets that are forwarded from segment 2 onto segment 9.

```
5:PowerHub:mgmt# pm viewpair 2 9 on 1
2->9 being viewed on 1
```

The next example sets up segment 10 to monitor all packets that are forwarded from segment 6 onto segment 8, and all packets forwarded from segment 8 to segment 6.

```
5:PowerHub:mgmt# pm viewpair 6 8 on 10 r
6<->8 being viewed on 10
```

9.9.6 Using Multiple Port Monitoring Commands

You can use multiple **`port-monitor`** commands to establish complex Port Monitoring arrangements. If you issue multiple **`port-monitor view`** and **`port-monitor viewpair`** commands, their effects are cumulative. That is, the PowerHub 6000 monitors ***all*** of the traffic specified by ***all*** of the commands, with one exception. If a **`port-monitor view`** command specifies the same option (incoming, forwarded, or generated) on the same monitoring segment as a previous **`port-monitor view`** command, then the new *`<monitored-seg-list>`* replaces the one given in the previous command(s).

Here are some examples of how multiple **`port-monitor`** commands can be used. In the following example, multiple commands set up segment 12 to monitor all packets that are generated by the PowerHub 6000 on segment 2, forwarded by the hub from segment 3 or 4 to segment 2, or received on segment 2.

```
8:PowerHub:mgmt# pm view g 2 on 12
Port 2 (g) being viewed on: 12
9:PowerHub:mgmt# pm viewpair 3 2 on 12
3->2 being viewed on 12
10:PowerHub:mgmt# pm viewpair 4 2 on 12
4->2 being viewed on 12
11:PowerHub:mgmt# pm view i 2 on 12
Port 2 (i) being viewed on: 12
```

The following command example shows how:

- Segment 6 is set up to monitor all packets that are forwarded by the PowerHub 7000 on segment 2, forwarded by the hub from segment 2 to segment 3 or 4, or forwarded in any direction between segments 3 and 4.
- Segment 7 is set up to monitor all packets that are generated by the PowerHub 7000 on segment 3 or 4.

```
12:PowerHub:mgmt# pm view f 2 on 6
Port 2 (f) being viewed on: 6
13:PowerHub:mgmt# pm viewpair 2 3 on 6
2->3 being viewed on 6
14:PowerHub:mgmt# pm viewpair 2 4 on 6
2->4 being viewed on 6
15:PowerHub:mgmt# pm viewpair 3 4 on 6 r
3<->4 being viewed on 6
16:PowerHub:mgmt# pm view g 3,4 on 7
Ports 3,4 (g) being viewed on: 7
```

9.9.7 Displaying the Current Monitoring State

To display the current Port Monitoring state, issue this command:

`port-monitor|pm`

This command displays a list of all currently-monitored segments and pairs.

9.9.8 Stopping Port Monitoring

To make the hub stop monitoring traffic, issue this command:

`port-monitor|pm close`

Issuing this command cancels the effect of all previous **`port-monitor view`** and **`port-monitor viewpair`** commands.

10 The Boot PROM Commands

This chapter describes the commands in the Packet Engine boot PROM. The boot PROM commands are accessible from the `<PROM-6pe>` prompt. The `<PROM-6pe>` prompt is not accessible during normal runtime operation, but is displayed if you interrupt the boot process or an error occurs during booting. (See Section 4.3.3 on page 59.)

Some of the boot PROM commands let you configure values in the Packet Engine's NVRAM (non-volatile RAM). These commands also are available in the **nvram** subsystem. The NVRAM settings remain in effect regardless of whether you set them from the **nvram** subsystem or from the `<PROM-6pe>` prompt.

You can use boot PROM commands to perform the following tasks:

- Boot the PowerHub software. (See Section 10.2 on page 205.)
- Display the MAC-layer hardware address of the Packet Engine. (See Section 10.3 on page 205.)
- Display a directory of the files contained on a floppy diskette or on the Flash Memory Module. (See Section 10.4 on page 205.)
- Display a file contained on a floppy diskette or in the Flash Memory Module. (See Section 10.5 on page 205.)
- Configure values in the Packet Engine's NVRAM. (See Section 10.6 on page 206.)
- Upload a file from a PC or Macintosh to the PowerHub floppy drive or Flash Memory Module. (See Section 10.7.1 on page 207.)
- Download a file from the PowerHub floppy drive or Flash Memory Module to a PC or Macintosh. (See Section 10.7.2 on page 209.)

> **NOTE**: To upload or download files between the PowerHub 6000 and a PC or Macintosh, the PC or Macintosh must support ZMODEM or XMODEM.

10.1 PACKET ENGINE BOOT PROM COMMANDS

Table 10–1 lists the commands in the Packet Engine boot PROM. No management capability is associated with these commands. For each command, this table lists the section in this chapter that contains more information about the command.

TABLE 10–1 Packet Engine boot PROM commands.

<table>
<tr><th>Command and Description</th><th>Capability*</th><th>See…</th></tr>
<tr><td>boot|b [fd|net|fm]
Boots the PowerHub software, using the device you specify as the boot source.</td><td>n/a</td><td>10.2</td></tr>
<tr><td>ethaddr|ea
Displays the MAC-layer hardware address of the Packet Engine (and therefore the system). Equivalent to the mgmt ethaddr command.</td><td>n/a</td><td>10.3</td></tr>
<tr><td>ls|dir [<file-spec> [<file-spec>...]]
Displays a directory of the files in the Flash Memory Module. Equivalent to the mgmt dir command.</td><td>n/a</td><td>10.4</td></tr>
<tr><td>more [-<rows>] <file-name> [<file-name>...]
Displays a file located in the Flash Memory Module. Similar to the mgmt showfile command, except more lets you specify how many lines to display at a time.</td><td>n/a</td><td>10.5</td></tr>
<tr><td>nvram [set|unset|show]
Sets, displays, or removes the setting from an NVRAM parameter. In the runtime software, you can display, set, or remove the setting from NVRAM parameters using commands in the nvram subsystem. See Chapter 11 for information about the NVRAM commands.</td><td>n/a</td><td>10.6</td></tr>
<tr><td>zreceive|zr|rz [-+27abcehtw] [<file-name>]
Uploads a file from a PC or Macintosh that supports ZMODEM or XMODEM onto the Flash Memory Module.</td><td>n/a</td><td>10.7.1</td></tr>
<tr><td>zsend|zs|sz [-+27abehkLlNnoptwXYy] <file-name>
Downloads a file from the Flash Memory Module onto a PC or Macintosh that supports ZMODEM or YMODEM.</td><td>n/a</td><td>10.7.2</td></tr>
<tr><td colspan="3">*R= Root, M=Monitor
R=display and manipulate, M=manipulate only.</td></tr>
</table>

10.2 BOOTING THE POWERHUB SOFTWARE

Use the **boot** command to boot the PowerHub software from the `<PROM-6pe>` prompt. Here is the syntax for this command:

boot|b [net|fm]

where:

net Boots the software over the network (the hub must be configured for netbooting).

fm Boots the software from the Flash Memory Module.

If you do not specify a boot source, the boot order configured in NVRAM is used. If you have not configured a boot order, the Flash Memory Module (**fm**) is used.

Note that the system software image file (6pe) must already be present on the boot source you specify. The **boot** command does not affect the boot order specified in NVRAM. (See Section 11.4 on page 214.)

10.3 DISPLAYING THE PACKET ENGINE'S HARDWARE ADDRESS

Use the **ethaddr|ea** command to display the MAC-layer hardware address of the Packet Engine. This command is equivalent to the **mgmt ethaddr** command. (See Section 9.3.5 on page 160.)

10.4 DISPLAYING A DIRECTORY

Use the **ls|dir** command to list a directory of the files in the Flash Memory Module. These commands are similar to the **mgmt dir** command. (Section 9.6.2 on page 177.) Each command displays the volume name, the files contained on the volume, and the amount of free space on the volume. Here is the syntax for these commands:

ls|dir [*<file-spec>* **[***<file-spec>...***]]**

<file-spec> Specifies a file name. You can use the wildcards (***** and **?**) for any portion of the file name. (See Section 9.6.2 on page 177.)

10.5 DISPLAYING A FILE

Use the **more** command to display a file located in the Flash Memory Module. Here is the syntax for this command:

more [-*<rows>***]** *<file-name>* **[***<file-name>...***]**

where:

-*<rows>* Specifies how many rows of the file are displayed at a time.

<file-name> Specifies a file name. The file is assumed to be located on the Flash Memory Module.

10.6 DISPLAYING, SETTING, OR UNSETTING AN NVRAM PARAMETER

Use the following **nvram** commands to display, set, or remove the setting from the following NVRAM parameters:

- Boot source order.
- Boot definition file to be used for local booting (Flash Memory Module) or netbooting (TFTP server).
- PowerHub IP address.
- PowerHub local subnet mask.
- Gateway address (when a gateway (router) separates the client hub from the BOOTP server).
- A TFTP server address.
- Post system-crash behavior.
- Segment allocations for chassis slots.

All parameters except the last two are used to configure boot parameters for the hub. The post-system crash behavior determines whether the hub attempts a reboot following a system crash. The segment allocations determine how many segments are allocated to each chassis slot.

The NVRAM commands in the Packet Engine boot PROM are identical to their counterparts in the **nvram** subsystem. The only difference is in whether you preface the command with "**nvram**":

- If you issue an NVRAM command from the `<PROM-6pe>` prompt, or from a subsystem other than **nvram**, preface the command with "**nvram**".
- If you issue the command from within the **nvram** subsystem, you do not need to preface the command with "**nvram**".

See Chapter 11 for a description of the NVRAM commands and parameters.

10.7 UPLOADING AND DOWNLOADING FILES

The PowerHub software supports ZMODEM, a file transfer protocol widely available for PCs and Macintoshes. Using the PowerHub ZMODEM commands, you can copy files between the PowerHub Flash Memory Module and a PC or Macintosh that is running a ZMODEM application. If your PC or Macintosh supports XMODEM, you can use the ZMODEM commands to perform XMODEM transfers.

To use the ZMODEM commands, a PC or Macintosh that supports ZMODEM must be connected to TTY1 or TTY2:

- If your management terminal on TTY1 contains the ZMODEM or XMODEM application, you can issue the PowerHub command to set up the transfer, then switch to your ZMODEM or XMODEM application to activate the transfer.
- Alternatively, you can set up the transfer from the management terminal on TTY1, then use the ZMODEM or XMODEM application on the device connected to TTY2 to complete the transfer.

10.7.1 Uploading a File to the PowerHub 6000

The **zreceive** command prepares the PowerHub software to receive a file to the Flash Memory Module. After you issue the **zreceive** command, you then must use the ZMODEM or XMODEM application on the PC or Macintosh to actually begin the transfer.

The **zreceive** command gives you numerous options, but the defaults for those options are appropriate for most file transfers. The defaults are the same for XMODEM transfers, except you must specify the file name when performing an XMODEM transfer. You do not specify the file name when performing a ZMODEM transfer.

Here is the syntax for the **zreceive** command:

zreceive|zr|rz [-+27abcehtw] [*<file-name>***]**

where:

-	Introduces the argument list. If you specify any arguments, the argument list must be preceded by the **-**. If you do not use any arguments, do not specify the **-**.
+	Causes the file you are receiving to be appended to an existing file. Specify the file name at the end of the argument list (ex: **zr -+ae hub1.log**).
2	Causes the file transfer to take place through the TTY2 port, rather than the TTY1 port. By default, the transfer takes place over the TTY1 port.
7	Uses 7-bit bytes for the transfer. By default the ZMODEM program uses 8-bit bytes.
a	Performs the transfer in ASCII mode, using the appropriate newline translation. By default, the transfer takes place in binary mode.
b	Performs the transfer in binary mode. Binary is the default transfer mode.

c	XMODEM only. Uses a 16-bit CRC.
e	Ignores control characters.
h	Sets the serial baud rate to 19.2 Kbps.
t	Sets the receive timeout to N/10 seconds (10 <= N <= 1000). Specify the file name at the end of the argument list (ex: **zr -at 500**). The default is **100**; that is, 10 seconds.
w	Sets the protocol window to N bytes. Specify the file name at the end of the argument list (ex: **zr -aw 10**).
<file-name>	Causes the XMODEM program to be used. Do not specify a file name if you intend to use ZMODEM. The PowerHub software assumes that you want to write the file to the Flash Memory Module.

If you specify none of the optional arguments nor a file name, the PowerHub software prepares to use the ZMODEM protocol to receive a file on TTY1, in binary mode, using 8-bit bytes. When you have issued the **zr** command on the hub, you then need to use the ZMODEM application on the PC or Macintosh to specify the file to be transferred and perform the transfer.

If you specify only a file name, the PowerHub software prepares to receive the named file using the XMODEM protocol, using the same default settings used for the ZMODEM protocol. After you issue the **zr** command and specify a file name, you then need to use the XMODEM application on the PC or Macintosh to perform the transfer.

Here are some examples of the use of this command. For each example, after issuing the command on the PowerHub 6000, you use the ZMODEM or XMODEM application on the PC or Macintosh to send the file.

In the first example, ZMODEM is used to prepare the PowerHub 6000 to receive a binary file from the PC or Macintosh. All the defaults are used. Notice that no file name is specified. When using ZMODEM, you specify the file name on the PC or Macintosh.

```
<PROM-6pe> zreceive
```

In the following example, the PowerHub 6000 is prepared to receive a configuration file from a PC or Macintosh. All defaults are accepted, except the default for transfer mode. The **a** argument is used to change the transfer mode to ASCII, which is appropriate for PowerHub configuration, environment, and boot definition files. Notice that the hyphen (**-**) is used to introduce the argument list.

```
<PROM-6pe> zreceive -a
```

The following example shows how the PowerHub system is prepared to upload a file using XMODEM. Because XMODEM, unlike ZMODEM, does not send file names from the sending device, the receiving device (the hub in this case) must supply the file name.

```
<PROM-6pe> zreceive 6pe
```

10.7.2 Downloading a File to a PC or Macintosh

Use the **zsend** command to download a file from the PowerHub 6000 to a PC or Macintosh. You can use this command to perform transfers to devices that support ZMODEM or YMODEM. Like the **zreceive** command, the **zsend** command provides many options, but the defaults for these options are appropriate for most transfers.

Here is the syntax for this command:

zsend|zs|sz [-+27abehkLlNnoptwXYy] *<file-name>*

where:

-	Introduces the argument list. If you specify any arguments, the argument list must be preceded by the **-**. If you do not use any arguments, do not specify the **-**.
+	ZMODEM only. Causes the file you are receiving to be appended to an existing file. Specify the file name at the end of the argument list (ex: **zs -+a hub2.log**).
2	Causes the file transfer to take place through the TTY2 port, rather than the TTY1 port. By default, the transfer takes place over the TTY1 port.
7	Uses 7-bit bytes for the transfer. By default the ZMODEM program uses 8-bit bytes.
a	Performs the transfer in ASCII mode, using the appropriate newline translation. By default, the transfer takes place in binary mode.
b	Performs the transfer in binary mode. Binary is the default transfer mode.
e	ZMODEM only. Escapes all control characters.
h	Sets the serial baud rate to 19.2 Kbps.
k	YMODEM only. Transfers the file in 1024-byte packets.
L	ZMODEM only. Limits the subpacket length to N bytes. Specify the file name at the end of the argument list (ex: **zs -aL 64**).
l	ZMODEM only. Limits the frame length to N bytes (l>=L). Specify the file name at the end of the argument list (ex: **zs -al 64**).
N	ZMODEM only. Transfers the file only if the version on the hub is both newer and longer than the version on the PC or Macintosh. Otherwise, the file is not transferred.
n	ZMODEM only. Similar to **N**, except the file is transferred if it is newer, even if it is not longer than the identically named file on the PC or Macintosh.
o	ZMODEM only. Uses a 16-bit CRC rather than the default 32-bit CRC.
p	ZMODEM only. Protects the file specified by *<file-name>* if it already exists on the PC or Macintosh. If the file is already present on the device, it is not overwritten by the file on the hub.
t	Sets the receive timeout to N/10 seconds (10 <= N <= 1000). Specify the file name at the end of the argument list (ex: **zs -at 500**). The default is **600**; that is, 60 seconds.

w	ZMODEM only. Sets the protocol window to N bytes. Specify the file name at the end of the argument list (ex: **zs -aw 10**).
X	Uses the XMODEM protocol, rather than the ZMODEM protocol. Note that some of the other arguments are not valid with the ZMODEM protocol.
Y	ZMODEM only. Overwrites the file specified by *<file-name>*, but skips the file altogether if the file is not present on the PC or Macintosh. If the file specified by *<file-name>* is not already present on the PC or Macintosh, the file is not written to that device.
y	ZMODEM only. Overwrites the file specified by *<file-name>*. Unlike **Y**, the **y** argument does not skip the file if it is not on the PC or Macintosh. Even if the file is not on the device, the hub writes the file to the device.
<file-name>	Specifies the file name you want to send. The PowerHub software assumes the file is located on the Flash Memory Module.

Here are some examples of the use of this command. In the first example, ZMODEM is used to send a software image file to the PC or Macintosh. All the **zsend** defaults are used.

```
<PROM-6pe> zsend fore/images/6pe
```

In the following example, an environment file is sent to the PC or Macintosh connected to TTY1. The **a** argument is used to change the transfer mode to ASCII, which is appropriate for PowerHub configuration, environment, and boot definition files. In addition, the **N** argument is used to ensure that the file is transferred only if it is newer and longer than the identically named file on the PC or Macintosh. Notice that the hyphen (**-**) is used to introduce the argument list and no spaces separate the arguments in the list. A space does separate the file name from the argument list.

```
<PROM-6pe> zsend -aN root.env
```

11 The NVRAM Subsystem

This chapter describes the NVRAM commands, located in the **nvram** subsystem. These commands let you configure values stored in the Packet Engine's NVRAM (Non-Volatile RAM). Using these commands, you can set, remove the setting from, or display the following values:

- Boot source order.
- Boot definition file to be used for local booting (Flash Memory Module) or netbooting (TFTP server).
- PowerHub IP address.
- PowerHub local subnet mask.
- IP address of the TFTP server.
- IP address of the gateway (when a gateway router separates the client hub from the BOOTP/TFTP server).
- Post system-crash behavior.
- Segment allocations on NIM slots (overrides default segment numbering).

The following sections describe how to display, set, or remove the setting for the NVRAM variables.

- To display the current setting of an NVRAM variable, see Section 11.3 on page 213.
- To set an NVRAM variable, see Section 11.4 on page 214.
- To remove an NVRAM setting, see Section 11.5 on page 216.

11.1 ACCESSING THE NVRAM SUBSYSTEM

To access the **nvram** subsystem, issue the following command at the runtime command prompt:

```
nvram
```

11.2 NVRAM SUBSYSTEM COMMANDS

Table 11–1 lists the **nvram** commands. For each command, the management capability (explained in Section 7.3 on page 125) is listed, as well as the section that contains additional information about the command.

TABLE 11–1 NVRAM commands.

Command and Description	Capability*	See…
set *<variable>* Sets a variable.	R	11.4
set *<variable> <value>* Sets a variable to the specified value.	R	11.4
show *<variable>* Displays the current settings for the specified variable.	R or M	11.3
unset *<variable>* Removes the setting from the specified variable.	R	11.5
***R= Root, M=Monitor.**		

11.3 DISPLAYING AN NVRAM VARIABLE

Use the following command to display the current setting of an NVRAM variable:

show *<variable>*

where:

<variable> Specifies one of the following:

bo The boot source order. You can specify one of the following:

- **m** Attempt to boot only from the Flash Memory Module. (This is the default.)
- **n** Attempt to boot only from the network.
- **mn** Attempt to boot from the Flash Memory Module first. If unsuccessful, boot from the network.
- **nm** Attempt to boot from the network first. If unsuccessful, boot from the Flash Memory Module.

locbdfile The boot definition file the hub uses when the software is booted from the Flash Memory Module.

netbdfile The boot definition file the hub uses when the software is booted from a BOOTP/TFTP server.

myip The PowerHub IP address.

mysm The PowerHub IP subnet mask.

fsip A TFTP server's IP address.

gwip An intervening router's (gateway's) IP address.

crashreboot

Whether the PowerHub 6000 automatically attempts a reboot following an unexpected system crash. The default is **set**, which causes the hub to attempt a reboot following a crash. We recommend that you do not change this setting unless advised to do so by FORE Systems TAC.

slotsegs[*<slot>***]**

The number of segments you have allocated to the slot specified by *<slot>*.

> **NOTE**: You must include the brackets around the slot number. They are part of the command you type.

Here are some examples of the use of this command:

```
25:PowerHub:nvram# show bo
bof (boot order: net)

26:PowerHub:nvram# show myip
myip 165.128.128.27
```

11.4 SETTING AN NVRAM VARIABLE

You can use the **nvram set** command to set the values used by the PowerHub 6000 for:

- Netbooting (see Section B.2.3 on page 235).
- Behavior following an unexpected system crash.
- The number of segments allocated to each NIM slot.

To set one of these NVRAM variables, issue the following command:

set *<variable> <value>*

where:

<variable> Specifies one of the following:

bo The boot source order. You can specify one of the following for *<value>*:

- **m** Attempt to boot only from the Flash Memory Module. (This is the default.)
- **n** Attempt to boot only from the network.
- **mn** Attempt to boot from the Flash Memory Module first. If unsuccessful, boot from the network.
- **nm** Attempt to boot from the network first. If unsuccessful, boot from the Flash Memory Module.

> **NOTE**: If you specify more than one boot source, make sure the configuration files on each source match. Otherwise, the hub's configuration might differ depending upon the boot source.

locbdfile The boot definition file the hub uses when the software is booted from the Flash Memory Module.

netbdfile The boot definition file the hub uses when the software is booted from a TFTP server.

myip The PowerHub IP address.

`mysm` The PowerHub IP subnet mask.

`fsip` A file server's IP address.

`gwip` An intervening router's (gateway's) IP address.

`crashreboot`

Whether the PowerHub 6000 automatically attempts a reboot following an unexpected system crash. The default is **`set`**, which causes the hub to attempt a reboot following a crash. We recommend that you do not change this setting unless advised to do so by FORE Systems TAC.

`slotsegs[`*`<slot>`***`]`** *<num>*

The number of segments you want to allocate to the slot specified by *`<slot>`*.

> **NOTE**: You must include the brackets around the slot number. They are part of the command you type.

Here are some examples of the use of this command. In the first example, values are configured into the hub's NVRAM to support "semi-prescient" netbooting.

```
27:PowerHub:nvram# set bo mn
28:PowerHub:nvram# set myip 147.128.16.1
29:PowerHub:nvram# set mysm 255.255.255.0
30:PowerHub:nvram# set fsip 147.128.16.2
31:PowerHub:nvram# set gwip 147.128.16.3
```

In the following example, the **`nvram set slotsegs`** command is used to override the default segment numbering for slot 2. This command ensures that the system allocates two segment numbers to slot 2, regardless of the presence or type of daughter card installed in slot 2.

Normally, the system numbers all the segments according to the connectors actually present in the chassis. (See Section 1.3.1 on page 18.) Accordingly, the segment numbers for the NIM in slot 2 depend on the presence and type of daughter card installed in slot 2. By overriding the default segment numbering, the command in this example ensures that the NIM in slot 3 always begins at segment number 15. (There are 12 segments allocated for slot 1 and 2 segments allocated for slot 2, totalling 14. The next segment number is 15.)

```
1:PowerHub:nvram# set slotsegs[2] 2
```

11.5 REMOVING THE SETTING FROM AN NVRAM VARIABLE

To remove the setting from an NVRAM variable, issue the following command:

unset *<variable>*

where:

<variable>	Is any one of the variables listed for the **set** command in Section 11.4.

Here is an example of the use of this command:

```
29:PowerHub:nvram# unset myip
```

Part 4: Appendices

This part contains the following appendices:

Appendix A:	Self-Tests	Describes how to run and interpret the loopback self-tests.
Appendix B:	Netboot Options	Describes the different types of netbooting implementations. In addition, this appendix describes boot definition files, macros, parsers and sharing methods.
Appendix C:	Pinouts	Lists the pinouts (pin signals) for 10Base-T, 100Base-TX, and 100Base-T4 segments.

Appendix A: Self-Tests

The PowerHub 6000 is shipped with test scripts that let you verify the complete functional operation of the system at maximum performance. These tests can help you verify that the hub was not damaged during shipping. The self-tests can simultaneously test both Ethernet and FDDI operation. Self-tests should be run only when the hub is not carrying live network traffic.

The PowerHub system software is shipped with the following self-test scripts:

`intloop` Configures an internal loopback test. This test checks the majority of a module's digital circuits, excluding the module's on-board transceivers that drive and receive packets on the segment connectors.[1] Packets are transmitted, looped back internally on the Ethernet chips, and received on the same segment that they were transmitted on.

`extloop` Configures an external loopback test. In addition to the circuitry checked by the internal loopback test, this test checks the on-board transceivers that drive and receive packets on the segment connectors. Packets are transmitted on one segment and received on another segment. To run this test, you need to remove your network cables and attach loopback cables between pairs of segments to be tested.

Each test script contains a series of configuration commands that set up the PowerHub 6000 for a test. To run a self-test, you perform the following steps:

- Execute the test script.
- Use the commands described in this appendix to inject one or more test packets into each segment to be tested.

1. For the Packet Engine, this test concentrates on the digital circuits used by the 10 Mb/s segments (slot 1) and by the daughter card (slot 2).

The following sections describe how to set up and run each test. When you finish running a self-test, always reboot the PowerHub 6000.

> **CAUTION**: Do not conduct loopback tests in a live network. To accomplish loopback, the tests put the bridging software into a special mode in which packets received on a particular segment are forwarded on the same segment, which is prohibited during normal operation. After running a loopback test, always reset the hub.

A.1 RUNNING THE INTERNAL LOOPBACK TEST

The internal loopback test checks most of the digital circuitry on a NIM, including the network interface chips. In addition, the internal loopback tests the 10 Mb/s circuitry and 100 Mb/s circuitry on the Packet Engine. It does not check the internal components of EMAs (Ethernet Media Adapters), nor does it check AUI Media Cables. It performs the test by placing each network interface chip into a mode where every packet that is transmitted on a particular segment is internally looped back and received on the same segment. For example, during the test, segment 1 both sends and receives the same packet continuously.

A.1.1 Setting up the Test

To run the internal loopback test:

(1) If the PowerHub 6000 is connected to a live network, disconnect the network cables, making a note of the segments to which they are attached.

(2) Reboot the PowerHub 6000.

(3) If the `login:` prompt is displayed, enter “`root`”. When the `password:` prompt is displayed, enter the password, if you have defined one. Otherwise, issue a carriage return (press Return or Enter).

(4) Issue the following command, then issue a carriage return.

```
mgmt rdcfg intloop
```

As soon as you issue a carriage return, the PowerHub software begins reading and executing the commands in the script file named `intloop`. As the system reads the commands in the script, it echoes them to the screen. You do not need to read or otherwise interpret these commands; they are not test results.

The commands in the `intloop` file attempt to set up 64 segments for loopback. (This allows for the fullest possible configuration.) If not all segments are present, you will see error messages as the script attempts to configure nonexistent segments. Ignore these error messages.

After the commands in `intloop` are executed, you can inject one or more packets into a loop as explained in Section A.1.2. The `intloop` script configures the hub so that when an injected packet is looped back and received, it is immediately retransmitted back out the same segment. Thus, the same packet continues to loop forever, unless it is dropped due to a hardware error.

> **NOTE**: No `cfg` file (default configuration file) can be present on the Flash Memory Module when you run the `extloop` test. Activities enabled by non-default configurations (such as the generation of Spanning-Tree and RIP packets) can interfere with the loopback test. This means you must perform these tests before using the **`mgmt savecfg cfg`** command to save your configuration. If you have saved the `cfg` file to the Flash Memory Module, use the **`mgmt rename`** *`<file-name>`* command to rename the `cfg` file.

A.1.2 Starting the Test

Use the **`diag x`** command to inject packets into a loop. Here is the syntax for this command:

`diag x` *`<seg-list>`***`|all`** *`<length>`* *`<num-pkts>`*

where:

*`<seg-list>`***`|all`**	Specifies the segments you want to test. You can specify a single segment, a comma-separated list of segments, or a hyphen-separated range of segments. If you specify **`all`**, all segments are tested.
`<length>`	Specifies the length of the packet(s) to be injected into the PowerHub 6000 for testing.
`<num-pkts>`	Specifies how many packets of the specified length you want to inject into the specified segments.

To start a loopback test on all segments in the system, issue the following command:

`diag x all 256 1`

This command injects one 256-byte packet on each looped-back segment and starts the loopback test.

> **NOTE**: For best results, limit the total number of packets sent out by a single **`diag x`** command to 30. That is, make sure the product of *`<num-pkts>`* and the number of segments in *`<seg-list>`* in a **`diag x`** command is 30 or less.

> **CAUTION**: The **diag** subsystem contains additional commands that are not intended for customer use. Do not attempt to use commands in the **diag** subsystem except as advised by FORE Systems TAC.

A.1.3 Increasing the Test Rate

It is possible to inject more than one packet into each loop. The PowerHub 6000 passes more traffic, as measured in both bits per second and packets per second, when there are multiple packets in each loop. This occurs because the hardware can send and receive a stream of packets back-to-back without waiting for the software when there are multiple packets waiting in the transmit queue.

You can add more packets to each loop simply by executing the **`diag x`** command multiple times, for example:

```
4:PowerHub:main# diag x all 256 1
5:PowerHub:main# diag x all 256 1
6:PowerHub:main# diag x all 256 1
7:PowerHub:main# diag x all 256 1
8:PowerHub:main#
```

The above commands inject four more 256-byte packets into each loop.[2] As you inject more packets, the performance of the PowerHub 6000, as measured by current utilization on the segments, continues to increase until you reach the processing limits of the Packet Engine for this test.

At this point, you may notice a slight irregular flickering of the LEDs on each segment. This occurs because the Packet Engine no longer has spare time to schedule packets for forwarding in a perfectly fair order, and from time to time some segments may forward fewer packets than others. Also at this point, the current-utilization statistics for different segments might be different.

> **NOTE**: When multiple packets are injected into the loop, you no longer can detect the loss of a single packet, because there are more packets in the loop to take its place.

A.1.4 Measuring Performance During a Test

You can determine the approximate performance of the PowerHub 6000 at any point during a loopback test by observing the number of packets transmitted during a fixed interval, using the following steps:

(1) Use a watch to measure a 10-second interval between entering the carriage returns in steps (2) and (3) below.

(2) Type the command **`bridge sc`** followed by a carriage return.

(3) Type the command **`bridge s all po`** followed by a carriage return.

(4) Calculate the number of packets per second (PPS) forwarded by the PowerHub 6000 as the sum of the numbers displayed in the previous step, divided by 10.

---

2. You can issue multiple **`diag x`** commands only for the internal loopback test. Do not issue multiple **`diag x`** commands for the external loopback test. If you do, the segments will experience excessive collisions and the test results will be invalid. You will need to re-start the test.

The amount of network bandwidth handled by the PowerHub 6000 in this test is calculated according to the following formula:

$$PPS \times bpp \times 2$$

where:

PPS Is packets-per-second.

bpp Is bits-per-packet.

This equation accounts for both transmit and receive bandwidth.

The number of bits per Ethernet packet can be calculated as follows:

$$bpp = (L + 20) \times 8$$

where L is the packet length in bytes. The additional 20 bytes account for the packet preamble and the interpacket gap.

For example, if 256-byte packets are used in the test, then the network bandwidth is PPS times 2208.

> **NOTE**: The accuracy of your measurements depends, of course, on how accurately you time the 10-second interval between commands. For better accuracy, you can measure the time over a longer interval, such as 100 seconds, then divide the sum in step (4) by the longer interval.

A.1.5 Interpreting the Test Results

The loopback test does not produce a report file or other formatted results. To assess whether the PowerHub 6000 is passing or failing the test, observe the segment LEDs. When the test is running, all the green activity (A/R) LEDs for the segments being tested should glow steadily. If all these LEDs begin to glow as soon as you begin the test and continue to glow steadily without faltering until you end the test, the PowerHub 6000 has passed this test.

> **NOTES**: The traffic LEDs are configurable. If you have used the **`mgmt led-config`** command to change the setting for the traffic LEDs from A and C to X and R, then both the A/R and X/C LEDs light up during this test. See Section 2.3.11 on page 29 and Section 9.4.3 on page 164 for information.

You also can look at statistics on the management terminal attached to port TTY1 to verify correct operation of the test. In particular, the following command shows for each segment the number of packets that are successfully looped back and the current utilization of the segment:

```
bridge s all po,pi,cu
```

The loopback test is successful only if the one packet that has been injected into a loop is successfully transmitted, received, and looped back indefinitely. If this packet is lost at any time, then there is nothing left to loop back and activity on that segment ceases. The transmit and receive LEDs go dark, and the packet statistics for that segment stop increasing.

Thus, if any of the transmit or receive LEDs fail to glow, or if they flicker or hesitate to light immediately, stop the test as described in Section A.1.6, then repeat the test. If you still observe dark, lagging, or flickering transmit or receive LEDs during the second run of the test, the PowerHub 6000 might be damaged. Contact FORE Systems, TAC or your service representative for assistance.

A.1.6 Stopping the Test

The internal loopback test continues to run indefinitely until you stop it. One way to stop the test is to reboot the hub by pressing the reset switch (marked `RST`) located on the Packet Engine, to the right of the `TTY1` port.

Another way to stop the test is to issue the **`bridge po all dis`** command. In this case, you then can start another test by issuing **`bridge po all enl`** followed by more **`diag x`** commands. You do not need to read the `intloop` file again.

> **CAUTION**: After running loopback tests, always reset the PowerHub 6000 before reconnecting it to your live network.

A.2 RUNNING THE EXTERNAL LOOPBACK TEST

The external loopback test checks the segment connection hardware, including the EMAs installed in the UMM (if present) and the 100 Mb/s daughter card, by sending a packet out one segment to another segment in a continuous loop. The segments are "paired" as one transmit segment and one receive segment. Each pair is connected by a loopback cable. Depending on how many loopback cables you have, you can create as many pairs as you like, provided the pairs are of similar media—10 Mb/s UTP segments must be attached to other 10 Mb/s UTP segments, MAUs connected to AUIs, and so on.

During the test, one or more packets are sent by each transmitting segment through the loopback cable to the receiving segment to which it is attached. The configuration established by the `extloop` script ensures that regardless of which segment a packet is received on, it is bridged and transmitted back out the same segment on which it was originally transmitted.

A.2.1 Setting up a Test

For this test, you need one or more loopback cables (one cable for each pair of segments). Loopback cables can be purchased from FORE Systems and other sources.

To set up the external loopback test:

(1) If some or all of the segments are attached to network cables, note what cables are attached to what segments, then disconnect the cables.

(2) Connect pairs of segments together with loopback cables. When looping BNC segments, be sure to set the termination switch at each end to the "`T`" (terminated) position.

(3) Reboot the PowerHub 6000.

(4) If the `login:` prompt is displayed, enter root. When the `password:` prompt is displayed, enter the password, if you have defined one. Otherwise, issue a carriage return (press Return or Enter).

(5) Issue the following command, followed by a carriage return:

`mgmt rdcfg extloop`

As soon as you issue the carriage return, the PowerHub software begins reading and executing the commands in the script file named `extloop`. As the system reads the commands in the script, it echoes them to the screen. You do not need to read or otherwise interpret these commands; they are not test results.

The commands in the `extloop` file attempt to set up 64 segments for loopback. (This allows for the fullest possible configuration.) If not all segments are present, you will see error messages as the script attempts to configure nonexistent segments. Ignore these error messages.

After the commands in `extloop` are executed, you can inject one or more packets into a loop as explained in Section A.2.2. The `extloop` configuration creates the following environment: when an injected packet is transmitted out one segment and received on another, it is immediately retransmitted back out the original segment. Thus, the same packet will keep looping forever, unless it is dropped due to a hardware error inside the PowerHub 6000 or a bit error on the loopback cable.

> **NOTE**: No `cfg` file (default configuration file) can be present on the Flash Memory Module when you run the `extloop` test. Activities enabled by non-default configurations (such as the generation of Spanning-Tree and RIP packets) can interfere with the loopback test. This means you must perform these tests before using the **`mgmt savecfg cfg`** command to save your configuration. If you have saved the `cfg` file to the Flash Memory Module, use the **`mgmt rename`** *`<file-name>`* command to rename the `cfg` file.

A.2.2 Starting the Test

As in the internal loopback test, you use the **`diag x`** command to inject packets into a loop. This command has the following format:

`diag x` *`<seg-list>`***`|all`** *`<length>`* *`<num-pkts>`*

where:

*`<seg-list>`***`|all`**	Specifies the segments you want to test. You can specify a single segment, a comma-separated list of segments, or even a hyphen-separated range of segments. If you specify **`all`**, all segments are tested.
`<length>`	Specifies the length of the packet(s) to be injected into the PowerHub 6000 for testing.
`<num-pkts>`	Specifies how many packets of the specified length you want to inject into the specified segments.

NOTE: For best results, limit the total number of packets sent out by a single **`diag x`** command to 30. That is, make sure the product of <*num-pkts*> and the number of segments in <*seg-list*> in any one **`diag x`** command is 30 or less.

CAUTION: The **diag** subsystem contains additional commands that are not intended for customer use. Do not attempt to use commands in the **diag** subsystem except as advised by FORE Systems TAC.

In the simplest application of the external loopback test, you send packets in only one direction on each loop. Therefore, for each loop, you must pick one segment to be the transmitting segment and the other to be the receiving segment, and transmit only on the transmitting segments.

For example, suppose you want to test the twelve UTP Segments on the Packet Engine. You could attach the loopback cables as shown in Figure A–1. You then could decide to transmit on all the odd-numbered segments by issuing the following command:

```
diag x 1,3,5,7,9,11 256 1
```

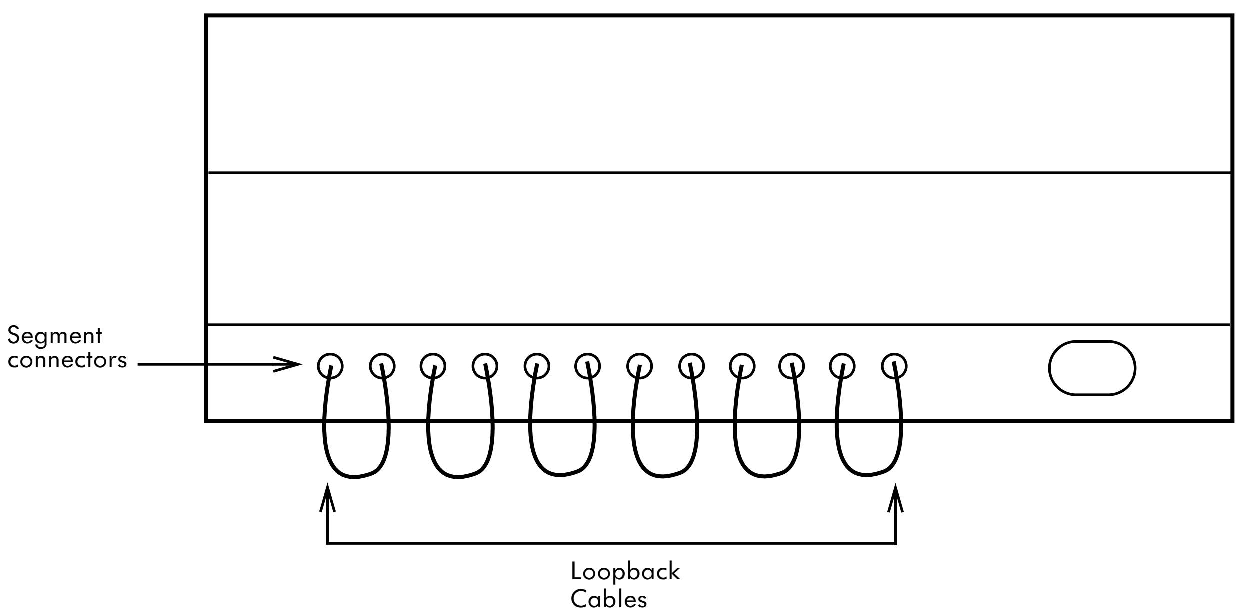


FIGURE A–1 Possible setup for external loopback test—UTP segments.

Alternatively, if the UUM is installed, bypassing the first six UTP segments in the Packet Engine, you could attach the loopback cables as shown in Figure A–2.

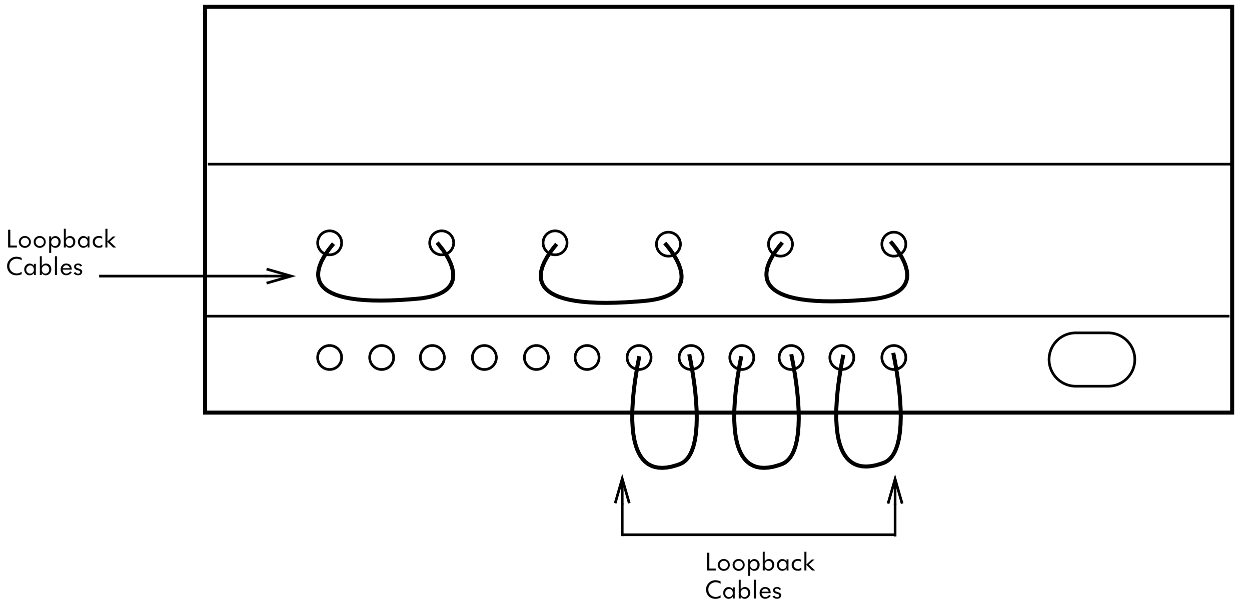


FIGURE A–2 Possible setup for external loopback test—chassis with UMM.

A.2.3 Interpreting the Test Results

Like the internal loopback test, the external loopback test does not produce a report file or other formatted results. To assess whether the PowerHub 6000 is passing or failing this test, observe the segment LEDs. Recall that each transmitting segment repeatedly sends one or more packets to a receiving segment. When a transmitting segment sends a packet, its transmit (X) LED flashes. When a receiving segment receives a packet, its receive (R) LED flashes.

Observe the LEDs for a few moments. If the transmit and receive LEDs for the corresponding segments are on, the PowerHub 6000 is passing the test.

However, if any of the transmit or receive LEDs fail to glow, or if you notice strange fluctuations in the pattern, end the test, then repeat the test. If you see the symptoms again, the PowerHub 6000 might be damaged. Contact FORE Systems TAC or your service representative for assistance.

A.2.4 Testing Both Directions

Referring again to Figure A–1, note that the loopback test that is started by the **`diag x`** command in Section A.2.2 checks the transmit circuitry on the odd-numbered segments and the receive circuitry on the even-numbered segments. To check the transmit and receive circuitry in the opposite direction, you first must stop the test as described in Section A.2.8, then restart the test by transmitting on the other segment of each pair. In the example test setup shown in Figure A–1, you would issue the following command:

```
diag x 2,4,6,8,10,12 256 1
```

A.2.5 Testing a Subset of the Segments

It is not necessary to perform the loopback test on all segments simultaneously. For example, if you have only a limited number of loopback cables available, you can start a test by only transmitting on one segment of each pair that actually has a loopback cable installed.

A.2.6 Bidirectional Testing

When looping back BNC, 10Base-FL, 10Base-FB, or UTP segments, it is possible to simultaneously transmit and receive in both directions. Collisions can occur in this case, but the bidirectional traffic usually “synchronizes” so that transmission and reception are interleaved on the loopback cable.

FORE Systems-supplied AUI loopback cables do not support bidirectional testing, because they do not deal with the collision signal. However, bidirectional testing is possible if the AUI segments are connected instead through a pair of external UTP, BNC, or 10Base-FL MAUs.

> **CAUTION**: In bidirectional testing, the traffic on a pair of segments can synchronize in an undesirable way. If both segments repeatedly begin their transmissions in perfect synchrony, they will experience multiple collisions. When 15 collisions occur in a row, the packet is dropped in accordance with the 802.3 Ethernet specification. The result is that transmission in one or both directions might cease. This behavior is especially likely if you inject multiple packets into each loop by issuing multiple **`diag x`** commands.

A.2.7 Full-Duplex Testing

You can perform full-duplex bidirectional testing on pairs of 10Base-FL or private UTP segments. To set up a full-duplex test, after reading in the `extloop` file use the **`mgmt operating-mode`** command to configure the desired segments for full-duplex operation:

`mgmt operating-mode` *`<seg-list>`* **`fdx`**

In the above command, *`<seg-list>`* contains both segment numbers of each pair that is to undergo full-duplex testing. You start a bidirectional test using the **`diag x`** command to inject a packet into both segments of each pair. No collisions can occur in full-duplex operation, so you can expect this test to run without dropping a packet until the test is terminated.

A.2.8 Stopping the Test

The external loopback test continues to run indefinitely until you stop it. One way to stop the test is to reboot the PowerHub 6000 by pressing the reset switch (marked `RST`) located on the Packet Engine, to the right of the `TTY1` port.

Another way to stop the test is to unplug, then re-insert, the loopback cables. If you do not want to unplug and re-insert the loopback cables, you can stop the test by issuing the **`bridge po all dis`** command. In this case, you then can start another test by issuing the **`bridge po all enl`** command followed by more **`diag x`** commands. You do not need to read the `extloop` file again.

> **CAUTION**: After running loopback tests, always reset the PowerHub 6000 before reconnecting it to your live network.

A.3 RESUMING NORMAL OPERATION

To begin or resume normal operation of the PowerHub 6000:

(1) If you ran the external test, unplug all the loopback cables and store them in a safe place.

(2) If you removed the cables for any of your real network segments, reconnect them to the segments.

(3) If you renamed the `cfg` (configuration) file, name it `cfg` again by issuing the following command:

`mgmt rename` *`<file-name>`* **`cfg`**

where:

`<file-name>` Is the name you gave the file in place of `cfg`.

(4) Reset the hub. (See Section 7.2 on page 124.)

Appendix B: Netboot Options

Section 4.3.4 on page 60 describes how to configure a single PowerHub 6000 to boot from a BOOTP/TFTP server. This appendix describes the netbooting process in detail and describes how you can share a common boot definition file among multiple PowerHub 6000s.

The PowerHub 6000 implementation of netbooting uses the Boot Protocol (BOOTP) and the TFTP (Trivial File Transfer Protocol). PowerHub netbooting is designed to be fully compliant with RFCs 951, 1048, and 1350. The PowerHub 6000 can netboot over any type of Ethernet or Fast Ethernet segment, but not over FDDI.

After you configure the PowerHub 6000 for netbooting, you can begin the netboot process by booting (or rebooting) the system, using any of the following methods:

- Press the reset switch (labeled `RST`), located on the front of the Packet Engine.
- Issue the **`mgmt reboot`** command. (See Section 9.3.1 on page 157.)
- Issue the **`boot`** (**`b`**) command at the `<PROM-6pe>` prompt. (See Section 10.2 on page 205.)
- Turn the power supply off, then back on.

B.1 CHOOSING A NETBOOTING METHOD

The boot process differs depending upon whether the client PowerHub 6000 and server are on the same subnet or different subnets. Accordingly, the netbooting implementation you need depends upon your network configuration.

You can use *point-to-point* netbooting if the client PowerHub 6000 and the BOOTP/TFTP server are on the same subnet. The subnet can be a single segment or multiple segments connected by bridges. Point-to-point netbooting is the simplest to implement. We recommend that you use point-to-point netbooting when the client PowerHub 6000 and the BOOTP/TFTP server are all on the same subnet. See Section 4.3.4.3 on page 62 for procedures.

If the client PowerHub 6000 and server are on different subnets, you can do one of the following:

- Implement a boot helper service on the client PowerHub subnet. The PowerHub software provides a service called *IP Helper* that can forward UDP packets (including BOOTP packets) between a netboot client and a remote server. See Section 4.3.4.4 on page 64 for procedures.
- Manually insert information (such as the client and server IP addresses and the IP address of the gateway) into the client PowerHub 6000's non-volatile RAM (NVRAM). The NVRAM contains a battery backup and retains its data across power cycles. See Section 4.3.4.4 on page 64 for procedures.

The boot parameters you can configure in NVRAM override the corresponding parameters returned by the BOOTP server. If you want to the hub bypass the BOOTP process, you can configure all the applicable boot values in NVRAM. See Section 11.4 on page 214 for details.

B.2 THE BOOT PROCESS

The netbooting process takes place in the following phases:

BOOTP	BOOTP packets are exchanged. (The BOOTP phase is bypassed if you configure all applicable boot parameters in the PowerHub 6000's NVRAM.)
BOOTDEF	Boot definition file is received via TFTP from server and parsed. The boot definition file specifies the configuration file and system software to be used.
IMAGE	Image files (system software) are received via TFTP from server and executed.
CONFIG	Configuration file is received via TFTP from server and executed.
RUN-TIME	Normal run-time operation begins.

The last four phases are identical for each netbooting implementation. However, the first phase (BOOTP) differs according to the implementation. For reference, the tables in the following sections summarize the netbooting process used by each method of netbooting. You do not need to know the netbooting phases in detail to implement netbooting, but these tables can help if you need to troubleshoot your netbooting implementation.

B.2.1 Point-to-Point

Table B–1 summarizes the netbooting process used when the client hub and the BOOTP server are on the same subnet.

TABLE B–1 Point-to-point netbooting.

Phase	Process
BOOTP	• Client hub (PowerHub 6000) sends a BOOTP broadcast packet out each Ethernet segment. The BOOTP packet contains client hub's MAC address, but no other address information. • Server receives broadcast packet and sends BOOTP reply packet to client hub (provided the hub's MAC address in the `booptab` file, or equivalent, on the BOOTP server). Reply packet contains server's IP address, client hub's IP address, client hub's IP subnet mask, and name of boot definition file. • Client hub receives BOOTP response and stores information from server in memory. **NOTE:** During the boot process, each Ethernet segment is configured as an IP interface by default. This segment configuration has no relation to the configuration of the segments during run-time operation.
BOOTDEF	• Client hub uses TFTP to transfer boot definition file from server. • Client hub parses boot definition file. • While parsing boot definition file, client hub obtains names of image files and configuration file and stores the names in memory.
IMAGE	• Client hub uses TFTP to transfer image files from server and loads them into memory, then executes them.
CONFIG	• Client hub gets, from memory, the name of the configuration file. • Client hub recalls which interface (segment) received BOOTP reply from server and configures that interface for TFTP exchanges. • Client hub uses TFTP to transfer configuration file from server and saves file in memory. • Client hub deconfigures interface. • Client hub executes configuration file.
RUN-TIME	• Client hub begins normal bridging and routing according to settings in configuration file.

B.2.2 Cross-Gateway (Boot Helper Service Used)

Table B–2 summarizes the netbooting process used when the client hub and BOOTP server are on separate subnets, and a boot helper service (such as IP Helper) is implemented. The intervening gateway that connects the segments can be another PowerHub Intelligent Switching Hub or any other device that implements a boot helper service.

TABLE B–2 Helper-assisted netbooting.

Phase	Process
BOOTP	• Client hub (PowerHub 6000) sends a BOOTP broadcast packet out each Ethernet segment. The BOOTP packet contains client hub's MAC address, but no other address information. • BOOTP request is received by intervening gateway on a segment previously configured with an IP Helper address. • IP Helper facility in intervening gateway forwards BOOTP packet to server. • Server receives BOOTP request forwarded by intervening gateway and sends response packet to gateway. Response packet contains name of boot definition file, server's IP address, client hub's IP address, client hub's IP subnet mask, and intervening gateway's IP address. • Gateway forwards response packet to client hub. • Client hub receives BOOTP response and stores information from server in memory. **NOTE:** During the boot process, each Ethernet segment is configured as an IP interface by default. This segment configuration has no relation to the configuration of the segments during run-time operation.
BOOTDEF	• Identical to point-to-point process.
IMAGE	• Identical to point-to-point process.
CONFIG	• Identical to point-to-point process.
RUN-TIME	• Identical to point-to-point process.

B.2.3 Cross-Gateway (No Boot Helper Service Used)

Table B–3 summarizes the netbooting process used for cross-gateway netbooting when the gateway does not have a boot helper service. If you prefer, you can implement this method even if the intervening gateway does contain a boot helper service.

TABLE B–3 Cross-gateway netbooting—no boot helper service used.

Phase	Process
BOOTP	• Client hub uses boot parameters in NVRAM as substitute for BOOTP parameters. The following parameters can be specified in NVRAM: client hub's IP address and subnet mask, gateway's IP address, server's IP address, name of the client hub's boot definition file. The boot definition file contains the file names and pathnames of the software image files and configuration file. Unless all BOOTP parameters were supplied from NVRAM, client hub (PowerHub 6000) sends a BOOTP broadcast packet out each Ethernet segment configured as IP interface. Parameters not configured in NVRAM are sought in the response from the BOOTP server. • BOOTP request is received by intervening gateway. If boot parameters in NVRAM include information needed by gateway to forward the BOOTP packet, the packet is forwarded to server. This information includes the client hub's IP address and subnet mask and the server's IP address. • Server receives BOOTP request forwarded by intervening gateway and sends response packet, through the gateway, to the client hub. • Client hub receives BOOTP response and stores information from server, including name of boot definition file, in boot PROM. **NOTE:** During the boot process, each Ethernet segment is configured as an IP interface by default. This segment configuration has no relation to the configuration of the segments during run-time operation.
BOOTDEF	• Identical to point-to-point process.
IMAGE	• Identical to point-to-point process.
CONFIG	• Identical to point-to-point process.
RUN-TIME	• Identical to point-to-point process.

B.3 CONFIGURATION NOTES

This section describes the configuration requirements for your BOOTP server, TFTP file server, and client PowerHub 6000 for point-to-point netbooting. Implement this type of netbooting if your client hub, BOOTP server, and TFTP server are all attached to the same subnet. The subnet can be a single segment or multiple segments connected by bridges.

B.3.1 TFTP Server

Regardless of the netbooting method you choose to implement, you must perform the following configuration tasks for the TFTP server (even if the BOOTP server and TFTP server are the same device):

- Install the system software image files.
- Edit and install the boot definition file(s). You can install a separate boot definition file for each client hub or you can use boot definition macros to share a single boot definition file among multiple client PowerHub systems. (See Section B.4.1 on page 239.)
- For each client PowerHub system, install its configuration file.

You can install these files in the TFTP home directory or set up subdirectories. If you set up subdirectories, make sure you specify the pathnames in the boot definition file when you edit the file.

B.3.2 BOOTP Server

You can configure the same host device as both a BOOTP server and a TFTP server, or you can configure separate BOOTP and TFTP servers.

> **NOTE:** Although BOOTP and TFTP services can be provided by different hosts, using the same host results in faster booting because the client hub does not need to search across its interfaces multiple times for a server. In fact, some BOOTP servers do not support the file service from another host, so in such cases you do not have a choice.

Unless you configure all the required values into NVRAM, you must configure your BOOTP server to provide the following information to the client hub (even if the BOOTP server and TFTP server are the same device):

- Client hub’s IP address.
- Client hub’s subnet mask.
- Gateway’s IP address (if the client hub and server are on different subnets).
- TFTP server’s IP address.

- Name of the boot definition file (often called `bootdef`) you plan to use to boot the client hub. You install this file on the TFTP server, but specify the name on the BOOTP server or in NVRAM. Note that the boot definition file is neither the image file (`6pe`) nor a configuration file (such as `cfg`).

The procedures for configuring your BOOTP server depend upon the BOOTP software you are using. In some BOOTP software, a single database file contains the information items listed above for each client hub that will use the server. In some implementations, this file is called the `bootptab` file. See your BOOTP software documentation for information.

B.3.3 Intervening Gateway

If a gateway separates the client hub from the server, you must do one of the following:

- If the gateway has a boot helper service, such as IP Helper, configure the helper service to help BOOTP packets sent from the client hub to reach the BOOTP server. If the gateway is another PowerHub Intelligent Switching Hub, use the **`ip add-helper`** command. (See Chapter 5 in the *PowerHub Software Manual, V 2.6* (Rev C).)
- If the gateway does not have a boot helper service, configure the following values in the client hub's NVRAM:
  - Client hub's IP address.
  - Client hub's subnet mask.
  - Gateway's IP address (if the client hub and server are on different subnets).
  - TFTP server's IP address.
  - Name of the boot definition file.

B.3.4 Client PowerHub

To configure the hub for netbooting:

- Specify the boot order in NVRAM. You must do this regardless of the type of netbooting you implement.
- If needed, configure boot parameters in NVRAM. See the previous section.

B.4 BOOT DEFINITION FILES

A *boot definition file* contains instructions for loading the system software files and the configuration file used by a PowerHub 6000 when it boots. This section describes boot definition files and boot definition macros, then tells you how to edit and copy boot definition files and configuration files onto the TFTP server.

The TFTP server must contain at least one boot definition file. You can edit and install a separate boot definition file for each client hub, or you can share a single boot definition file among multiple client hubs. The software diskettes contain a boot definition file called `bootdef`. This `bootdef` file supports booting from the Flash Memory Module, but you can copy and modify the file for netbooting.

Here is an example of the `bootdef` file that is shipped with the PowerHub 6000:

```
%vstart 1
6pe             m
%vend 1
```

To prepare a `bootdef` file for netbooting, copy the file from one of the floppy diskettes shipped with the PowerHub system onto the TFTP server, then modify the file as follows:

- Add or modify a line to load the configuration file. (If you already saved the configuration file before copying the `bootdef` file, the `bootdef` file already contains a line for loading the configuration file. You need to modify the line.)
- Add the pathname and file name for the software image on the TFTP server.

Here is an example of a `bootdef` file that is modified for netbooting:

```
%vstart 1
fore/ph/configs/0000EF014A00.cfg     c
fore/ph/images/6-2.6.3.0/6pe         m
%vend 1
```

In this example, the client hub's MAC-layer hardware address is used as the configuration file name. The pathnames for the configuration file and the software image file are included with the file names. Whether you need to specify a pathname depends upon how your TFTP server is configured. When you edit the `bootdef` file, make sure you enter the pathnames that are meaningful to your TFTP server.

B.4.1 Using the Same Boot Definition File with Multiple Hubs

If you need to configure only one PowerHub system for netbooting, using the hub's MAC-layer hardware address to name the configuration file is a simple way to name the file. However, if you need to configure more than one hub for network booting, you might want to do one of the following:

- Create a unique boot definition file for each client hub. If you choose this method, you must use the **`nvram set netbdfile`** command to set the boot definition file in each client hub's NVRAM. Otherwise, each hub will attempt to use the default boot definition file name (`bootdef`). See Section 11.4 on page 214 for information about the **`nvram set netbdfile`** command.
- Use a single boot definition file, but use boot definition macros in place of the configuration file name. A *boot definition macro* is a 2-character sequence consisting of a '`$`' followed by a single letter. The macros are expanded by the client hub's Packet Engine boot PROM when it reads the boot definition file. Table B–4 lists the boot definition macros.

TABLE B–4 Boot definition macros.

Macro	Expands into...
`$E`	ASCII representation of the MAC-layer hardware address of the client hub; for example, "0000EF014A00". $E always expands to 12 characters.*
`$e`	ASCII representation of the three least significant octets of the MAC-layer hardware address of the client hub. `$e` always expands to 6 characters.*
`$D`	Directory part of the path name of the boot definition file.
`$B`	Base name of the boot definition file (the directory part of the path name and anything following the rightmost dot of the file name are removed).
`$$`	Expands to a single '`$`' character. Use this if you ever need to use the '`$`' character with a boot definition macro.
***$E and $e expand hex digits A-F in uppercase**	

Here is an example of a boot definition file that uses boot definition macros.

```
%vstart 1
$D/$E.cfg                               c
fore/ph/images/6-2.6.3.0/6pe            m
%vend 1
```

In this example, the `$D` macro expands into the pathname of the boot definition file. The `$E` macro expands into the client hub's MAC-layer hardware address. Assume that this boot definition file is used with the same client hub supported by the example boot definition file in Section B.4 on page 238. When that client hub's Packet Engine parses the macros in the boot definition file, it expands `$D` into `fore/ph/configs/` and `$E` into `0000EF014A00`.

Both the example boot definition file in Section B.4 on page 238 and the one shown above can be used with the same client hub. However, the first file (without boot definition macros) can be used only with a particular hub, whereas the second file (with boot definition macros) can be used with many client hubs.

B.4.2 Sharing Methods

If you choose to share a common boot definition file among multiple client hubs, you must decide on one of the following sharing methods:

MAC-address	Each configuration file is named according to the following convention: *`<MAC-addr>`*`.cfg`, where *`<MAC-addr>`* is the MAC-layer hardware address of the client hub.
Link	On boot servers that support symbolic links to files, you can give meaningful names to configuration files according to the following pattern: *`<name>`*`.cfg`, where *`<name>`* is an arbitrary name you assign to the client hub.

These methods are very similar. They differ only in that you can use any name for the configuration files if you use the link method. However, you must use the MAC-layer hardware addresses in the configuration file names if you use the MAC-address method.

> **NOTE**: To use the link method, your TFTP server must support symbolic links. To determine if your server supports symbolic links, see your server's documentation.

The following sections contain examples of each sharing method.

B.4.2.1 MAC-Address Method

Here is an example of a TFTP server directory and file structure used to implement the MAC-address sharing method.

```
fore

        ph
                images
                        6-2.6.3.0
                                6pe

                        6-2.6.3.1
                                6pe

                        6-2.6.3.2
                                6pe

                configs
                        bootdef-6-2.6.3.0
                        bootdef-6-2.6.3.1
                        bootdef-6-2.6.3.2

                        014A00.cfg
                        015AD0.cfg
                        015AE0.cfg
                        015AF0.cfg
                        016AD0.cfg
```

This example shows TFTP subdirectories, but you can just as easily install the files into the TFTP home directory. As shown in this example, the `configs` subdirectory contains a single boot definition file for each system software version, but a separate configuration file for each client hub that uses this TFTP server. Each configuration file is named after the last six hexadecimal digits of a client hub's MAC-layer hardware address.

Recall that the boot definition file contains the name of the configuration and system software image files to be loaded. The BOOTP server tells the client hub which boot definition file to use. The boot definition macros are expanded to form the unique name of the configuration file for that client.

To name your configuration files according to the MAC-address sharing method, use the following procedure for each client hub that will use the server.

For each configuration file, copy the file onto the server:
<MAC-addr>`.cfg`

where:

<MAC-addr>	Is the MAC-layer hardware address (in hex double-digit format) of the client hub. You can specify the full hardware address (12 hex digits) or, for systems that do not support long file names, the last three octets (six hex digits) of the hardware address. The extension "`cfg`" identifies the file as a configuration file.

NOTE: Hex digits A-F ***must*** be in uppercase.

When you complete this procedure, your TFTP server should contain a separate configuration file for each client hub that will use the server to netboot.

B.4.2.2 Link Method

Here is an example of a TFTP server directory and file structure used to implement the link sharing method.

```
fore

         ph
                 images
                         6-2.6.3.0
                                 6pe

                         6-2.6.3.1
                                 6pe

                         6-2.6.3.2
                                 6pe

                 configs
                         bootdef-6-2.6.3.0
                         bootdef-6-2.6.3.1
                         bootdef-6-2.6.3.2

                         ph-1.bd
                         ph-2.bd
                         ph-3.bd

                         ph-1.cfg
                         ph-2.cfg
                         ph-3.cfg
```

This example shows TFTP subdirectories, but you can just as easily install the files into the TFTP home directory. As shown in this example, the `configs` subdirectory contains a single boot definition file for each version of runtime software, and a configuration file for each client PowerHub that uses this TFTP server. It also contains a link to each configuration file. Each link has the same base name as the corresponding configuration file, but has the extension `.bd.` This base name is associated with a particular client hub by a `bootptab` file or other file containing IP information and other information needed by your BOOTP software. See the documentation for your file server to determine how to associate the link names with the configuration files.

Assume that a client hub's Packet Engine is parsing the example boot definition file shown in Section B.4. When it parses the configuration file line,

```
$D/$B.cfg             c
```

the file name expands into the directory of the configuration file (`$D`) and the base name (`$B`) of the boot definition file. This tells the client hub which directory and boot definition file on the TFTP server to use. Moreover, because a link has been defined between the boot definition file and a particular configuration file, that configuration file is used by the client hub.

To name your configuration files according to the link sharing method, use the following procedure for each client hub that will use the server.

For each configuration file, copy the file into the `/fore/ph/configs` directory as: `<name>.bd`

where:

<name>	Is a meaningful name that you give the file. You can use any file name that is legal on your server. The extension "`bd`" indicates that this is a symbolic file name.

When you complete this procedure, your TFTP server should contain a separate configuration file for each client hub that will use the server.

Appendix C: Pinouts

This appendix shows the pinouts for the 10Base-T (UTP), 100Base-TX, and 100Base-T4 segments.

As shown in Figure C-1, the RJ-45 connector has 8 pinouts. The 10Base-T connector uses only four of these pinouts. The 100Base-TX and 100Base-T4 connectors use all 8.

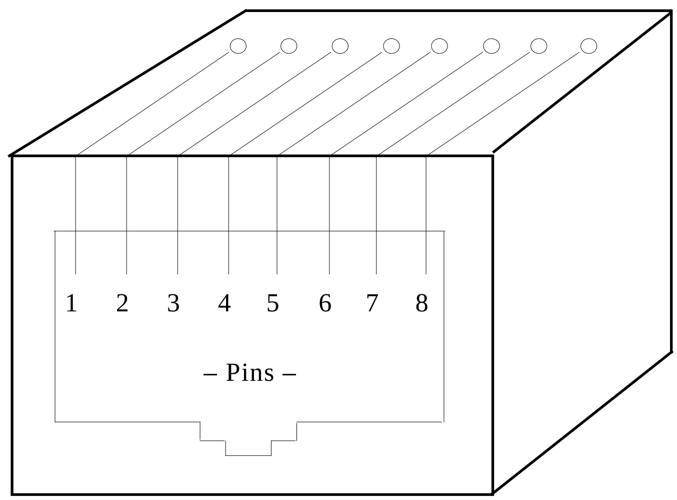


FIGURE C-1 Pin locations on RJ-45 connector.

The tables that follow list the pinouts for the specific media types that use the RJ-45 connector.

> **NOTE**: Although the pinouts used by 10Base-T and 100Base-TX are identical to each other, they are different from the standard used by TP-PMD, the FDDI-over-copper standard.

C.1 10BASE-T (UTP) PINOUTS

Table C–1 lists the pinouts for the 10Base-T segments using the RJ-45 connector. In this table, the signal names are defined with respect to a TP-MAU at the workstation or other network node. For example, the “transmit” signals (pins 1 and 2) are indeed transmit outputs at the TP-MAU, but they are receive inputs at the PowerHub 6000.

TABLE C–1 RJ-45 pinouts—10Base-T.

Pin No.	Signal Name	Hub	Workstation (TP-MAU)
1	MAU XMT POS	Input	Output
2	MAU XMT NEG	Input	Output
3	MAU RCV POS	Output	Input
4	unused	unused	n/a
5	unused	unused	n/a
6	MAU RCV NEG	Output	Input
7	unused	unused	n/a
8	unused	unused	n/a

C.2 100BASE-TX PINOUTS

Table C–2 lists the pinouts for the 100Base-TX segments using the RJ-45 connector. The signal names are defined with respect to a workstation with a DTE interface. For example, the “transmit” signals (pins 1 and 2) are indeed transmit outputs at the workstation, but they are receive inputs at the PowerHub 6000. Note that unlike the 10Base-T segments, pins 4, 5, 7, and 8 are used as pseudo-grounds to reduce EMI (electro-magnetic interference).

TABLE C–2 RJ-45 pinouts—100Base-TX.

Pin No.	Signal Name	Hub	Workstation (DTE)
1	MAU XMT POS	Input	Output
2	MAU XMT NEG	Input	Output
3	MAU RCV POS	Output	Input
4	Pseudo-ground	Pseudo-ground	Pseudo-ground
5	Pseudo-ground	Pseudo-ground	Pseudo-ground
6	MAU RCV NEG	Output	Input
7	Pseudo-ground	Pseudo-ground	Pseudo-ground
8	Pseudo-ground	Pseudo-ground	Pseudo-ground

C.3 100BASE-T4 PINOUTS

Table C–3 lists the pinouts for the 100Base-T4 segment on the RJ-45 connector. In this table, as in Table C–2, the signal names are defined with respect to a workstation with a DTE interface. For example, the “transmit” signals (pins 1 and 2) are indeed transmit outputs at the workstation, but they are receive inputs at the PowerHub 6000.

TABLE C–3 RJ-45 pinouts—100Base-T4 Connectors.

Pin No.	Signal Name	Hub	Workstation (DTE)
1	TX_D1+	Input	Output
2	TX_D1-	Input	Output
3	RX_D2+	Output	Input
4	BI_D3+	Input	Output
5	BI_D3-	Output	Input
6	RX_D2-	Output	Input
7	BI_D4+	Input	Output
8	BI_D4-	Output	Input

C.4 10BASE-T CHAMP PINOUTS

Figure C-2 shows the pin locations on each Champ connector. Note that the leftmost pins, 25 and 50, are not connected to a 10Base-T segment; they are unused.

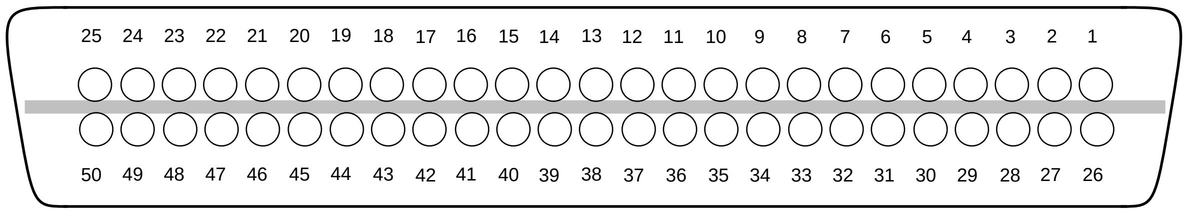


FIGURE C-2 Pin locations on Champ connector.

Figure C-3 shows the location of the pin signals for each segment on the Champ connector. Use this information along with the pinouts listed in Table C–4 to wire your network cables for the Champ segments. The segment numbers are for the Champ connector on the 12x1 NIM and the bottom Champ connector on the 24x1 NIM. The segment numbers in parentheses are for the top Champ on the 24x1.

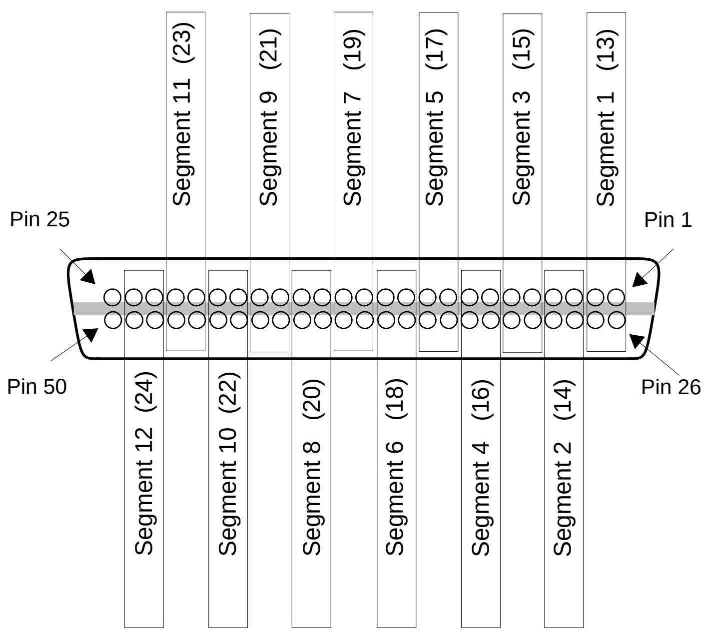


FIGURE C-3 Segment positions on Champ connector.

Table C–4 lists the pinouts for the Champ connectors. Segment numbers 1 through 12 are listed in the table. The pinout information also applies to segments 13 through 24. (See Figure C-3 on page 249.)

TABLE C–4 Champ connector pinouts.

Segment No.	Pin No.	Signal
1, 13	26	HUB RVC POS
	1	HUB RCV NEG
	27	HUB XMT POS
	2	HUB XMT NEG
2, 14	28	HUB RVC POS
	3	HUB RCV NEG
	29	HUB XMT POS
	4	HUB XMT NEG
3, 15	30	HUB RVC POS
	5	HUB RCV NEG
	31	HUB XMT POS
	6	HUB XMT NEG
4, 16	32	HUB RVC POS
	7	HUB RCV NEG
	33	HUB XMT POS
	8	HUB XMT NEG
5, 17	34	HUB RVC POS
	9	HUB RCV NEG
	35	HUB XMT POS
	10	HUB XMT NEG
6, 18	36	HUB RVC POS
	11	HUB RCV NEG
	37	HUB XMT POS
	12	HUB XMT NEG
7, 19	38	HUB RVC POS
	13	HUB RCV NEG
	39	HUB XMT POS
	14	HUB XMT NEG

TABLE C–4 (Continued) Champ connector pinouts.

Segment No.	Pin No.	Signal
8, 20	40	HUB RVC POS
	15	HUB RCV NEG
	41	HUB XMT POS
	16	HUB XMT NEG
9, 21	42	HUB RVC POS
	17	HUB RCV NEG
	43	HUB XMT POS
	18	HUB XMT NEG
10, 22	44	HUB RVC POS
	19	HUB RCV NEG
	45	HUB XMT POS
	20	HUB XMT NEG
11, 23	46	HUB RVC POS
	21	HUB RCV NEG
	47	HUB XMT POS
	22	HUB XMT NEG
12, 24	48	HUB RVC POS
	23	HUB RCV NEG
	49	HUB XMT POS
	24	HUB XMT NEG

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X